

AD-A184 057

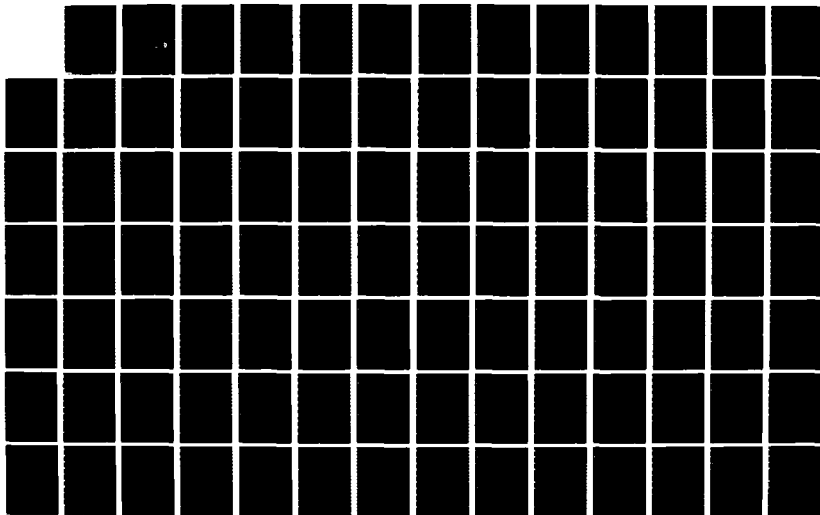
US ARMY INTELLIGENCE CENTER AND SCHOOL (USATCS)
SOFTWARE ANALYSIS AND MAN (U) JET PROPULSION LAB
PASADENA CA 8 PARD0 05 MAR 87 JPL-D-4216 NAS7-918

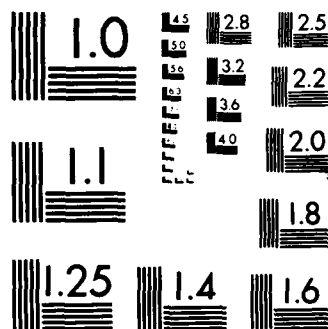
1/5

UNCLASSIFIED

F/G 12/5

ML





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

AD-A184 057

DTIC FILE COPY

2

ALGO_IGR_0049

U.S. ARMY INTELLIGENCE CENTER AND SCHOOL
(USAICS)

Software Analysis and Management System
(USAMS)

Improved GUARDRAIL V MC68000 'DF' Files

5-MAR-1987

Author:

Bruce Pardo

DTIC
SELECTED
AUG 26 1987
S D
C/D

JET PROPULSION LABORATORY

California Institute of Technology
Pasadena, California

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ALGO-IGR-0049	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Improved GUARDRAIL V MC68000 "DF" Files		5. TYPE OF REPORT & PERIOD COVERED Final
7. AUTHOR(s) B. Pardo		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Jet Propulsion Laboratory California Institute of Technology 4800 Oak Grove Drive, Pasadena, CA 91109		8. CONTRACT OR GRANT NUMBER(s) NAS7-918
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, USAICS ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-7000		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS RE 182 AMEND #187
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 5 Mar 87
		13. NUMBER OF PAGES 458
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NONE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Dissemination		
DISTRIBUTION STATEMENT A Approved for public release Distribution Unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Prepared by Jet Propulsion Laboratory for the US Army Intelligence Center and School's Combat Developer's Support Facility.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Assembly Code, MC68000 MPU, IGRV, DF, Fix Estimation, Computer Files, File Extensions ←		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report contains a collected list of source files containing direction finding (DF) and fix estimation software. This software is used in the Improved GUARDRAIL V (ICRV) system and is written in MC68000 microprocessor assembly code. All comments are included. Line numbers are added for easy reference. <i>Approved</i>		

ALGO_IGR_0049

**U.S. ARMY INTELLIGENCE CENTER AND SCHOOL
(USAICS)**

**Software Analysis and Management System
(USAMS)**

Improved GUARDRAIL V MC68000 'DF' Files

5-MAR-1987

Author:

Bruce Pardo

JET PROPULSION LABORATORY

**California Institute of Technology
Pasadena, California**

CONTENTS

1	INTRODUCTION	1
2	FILES WITH EXTENSION '.BAT'	2
2.1	DFALG.BAT_NUM	3
3	FILES WITH EXTENSION '.CMD'	4
3.1	DFALG.CMD_NUM	5
4	FILES WITH EXTENSION '.CNT'	6
4.1	DFALG.CNT_NUM	7
4.2	DFGRND.CNT_NUM	10
5	FILES WITH EXTENSION '.CSS'	11
5.1	DFOUT.CSS_NUM	12
6	FILES WITH EXTENSION '.D68'	13
6.1	DFBANDS.D68_NUM	14
6.2	DFCOL.D68_NUM	15
6.3	DFCOL2.D68_NUM	19
6.4	DFCOM.D68_NUM	23
7	FILES WITH EXTENSION '.DBS'	24
7.1	DF1.DBS_NUM	25
7.2	DFC.DBS_NUM	27
7.3	DFQ.DBS_NUM	28
8	FILES WITH EXTENSION '.DOC'	29
8.1	DFALG.DOC_NUM	30
8.2	DFGRND.DOC_NUM	31
9	FILES WITH EXTENSION '.EQU'	32
9.1	DFERRS.EQU_NUM	33
9.2	DFTYPE.EQU_NUM	34
10	FILES WITH EXTENSION '.INC'	35
10.1	DFERR.INC_NUM	36
10.2	DFTYPE.INC_NUM	37
11	FILES WITH EXTENSION '.MOD'	38
11.1	DFA1.MOD_NUM	39
11.2	DFA2.MOD_NUM	40
11.3	DFA4.MOD_NUM	41
11.4	DFA5.MOD_NUM	42
11.5	DFA6.MOD_NUM	43

11.6	DFG1.MOD_NUM	44
11.7	DFG2.MOD_NUM	45
11.8	DFG3.MOD_NUM	46
11.9	DFG5.MOD_NUM	47
11.10	DFG5T.MOD_NUM	48
11.11	DFG6.MOD_NUM	49
11.12	DFG7.MOD_NUM	50
11.13	DFG8.MOD_NUM	51
11.14	DFG9.MOD_NUM	52
11.15	DFIN.MOD_NUM	53
11.16	DFOUT.MOD_NUM	54
11.17	DFPE1.MOD_NUM	56
11.18	DFPE2.MOD_NUM	57
11.19	DFPON.MOD_NUM	58
12	FILES WITH EXTENSION '.MST'	59
12.1	DFALG.MST_NUM	60
12.2	DFGRND.MST_NUM	62
13	FILES WITH EXTENSION '.PAC'	64
13.1	DF.PAC_NUM	65
13.2	DFP.PAC_NUM	68
13.3	DFI.PAC_NUM	70
13.4	DFPR.PAC_NUM	71
13.5	DFR.PAC_NUM	73
13.6	DFRJ.PAC_NUM	75
14	FILES WITH EXTENSION '.R68'	76
14.1	DFA1.R68_NUM	77
14.2	DFA2.R68_NUM	78
14.3	DFA4.R68_NUM	82
14.4	DFA5.R68_NUM	83
14.5	DFA6.R68_NUM	85
14.6	DFBANDS.R68_NUM	88
14.7	DFCOL.R68_NUM	89
14.8	DFCOL2.R68_NUM	90
14.9	DFCOM.R68_NUM	91
14.10	DFFEED.R68_NUM	92
14.11	DFG1.R68_NUM	93
14.12	DFG2.R68_NUM	94
14.13	DFG3.R68_NUM	95
14.14	DFG5.R68_NUM	99
14.15	DFG5T.R68_NUM	100
14.16	DFG6.R68_NUM	101
14.17	DFG8.R68_NUM	102
14.18	DFIN.R68_NUM	103
14.19	DFOUT.R68_NUM	104



Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Date	
Availability Codes	
A-1	

14.20	DFPE1.R68_NUM	105
14.21	DFPE2.R68_NUM	106
14.22	DFPON.R68_NUM	108
14.23	DFRPT.R68_NUM	109
15	FILES WITH EXTENSION '.DEF'	112
15.1	DFSMENU1.DEF_NUM	113
15.2	DFSMENU2.DEF_NUM	117
15.3	DFSMENU3.DEF_NUM	120
15.4	DFSMENU4.DEF_NUM	121
15.5	DFSMENU5.DEF_NUM	123
15.6	DFSMENU6.DEF_NUM	127
15.7	DFSMENU7.DEF_NUM	129
15.8	DFSMENU8.DEF_NUM	133
16	FILES WITH EXTENSION '.FMT'	134
16.1	DFA1.FMT_NUM	135
16.2	DFA2.FMT_NUM	137
16.3	DFA3.FMT_NUM	140
16.4	DFA4.FMT_NUM	142
16.5	DFA5.FMT_NUM	144
16.6	DFA6.FMT_NUM	146
16.7	DFALG.FMT_NUM	148
16.8	DFG1.FMT_NUM	184
16.9	DFG2.FMT_NUM	186
16.10	DFG3.FMT_NUM	188
16.11	DFG5.FMT_NUM	190
16.12	DFG5T.FMT_NUM	192
16.13	DFG6.FMT_NUM	193
16.14	DFG7.FMT_NUM	195
16.15	DFG8.FMT_NUM	197
16.16	DFG9.FMT_NUM	199
16.17	DFGRND.FMT_NUM	201
16.18	DFIN.FMT_NUM	232
16.19	DFOUT.FMT_NUM	234
16.20	DFPE1.FMT_NUM	236
16.21	DFPE2.FMT_NUM	238
16.22	DFPON.FMT_NUM	240
17	FILES WITH EXTENSION '.LST'	241
17.1	DFA1.LST	242
17.2	DFA2.LST	253
17.3	DFA6.LST	285
18	FILES WITH EXTENSION '.PIC'	307
18.1	DF.PIC_NUM	308
18.2	DF1.PIC_NUM	311

18.3	DFC.PIC_NUM	313
18.4	DFF.PIC_NUM	314
18.5	DFPR.PIC_NUM	315
18.6	DFQ.PIC_NUM	318
18.7	DFR.PIC_NUM	319
18.8	DFRJ.PIC_NUM	322
19	FILES WITH EXTENSION '.S68'	323
19.1	DFA1.S68_NUM	324
19.2	DFA2.S68_NUM	328
19.3	DFA4.S68_NUM	350
19.4	DFA5.S68_NUM	352
19.5	DFA6.S68_NUM	357
19.6	DFFEED.S68_NUM	371
19.7	DFG1.S68_NUM	373
19.8	DFG2.S68_NUM	378
19.9	DFG3.S68_NUM	380
19.10	DFG5.S68_NUM	397
19.11	DFG5T.S68_NUM	401
19.12	DFG6.S68_NUM	403
19.13	DFG8.S68_NUM	406
19.14	DFIN.S68_NUM	410
19.15	DFOUT.S68_NUM	412
19.16	DFPE1.S68_NUM	416
19.17	DFPE2.S68_NUM	418
19.18	DFPON.S68_NUM	428
19.19	DFRPT.S68_NUM	431
20	FILES WITH EXTENSION '.SCR'	442
20.1	DFSMENU1.SCR_NUM	443
20.2	DFSMENU2.SCR_NUM	444
20.3	DFSMENU3.SCR_NUM	445
20.4	DFSMENU4.SCR_NUM	446
20.5	DFSMENU5.SCR_NUM	447
20.6	DFSMENU6.SCR_NUM	448
20.7	DFSMENU7.SCR_NUM	449
20.8	DFSMENU8.SCR_NUM	450

Improved GUARDRAIL V MC68000 'DF' Files

1 INTRODUCTION

This document comprises a collected arrangement of files that concern Direction-Finding and Fix Estimation. Improved GUARDRAIL V (also known as IGRV, IGR or AN/USD-9(V)2) is the Intelligence/Electronic Warfare (IEW) systems used for this analysis. Only files with the file name starting with "DF" are contained in this publication. The source code included is assembly code targetted for a Motorola MC68000 micro processor. The files are grouped by 80 character column width and 132 column width. Each of these sections inturn are arranged alphabetically. Note that some files presented here within contain text describing the function of the source code. This IEW system was developed by: Electromagnetic Systems Laboratory (ESL), Inc. ESL is located in Sunnyvale, California.

All source code embodied within this IEW system is UNCLASSIFIED. Only the parts of data referring to system limits and specific values which are processed by the source code are CLASSIFIED. No data of this latter type is included in this document. For more information on the security and privacy of the inclosed source code please direct attention to: IGRV REQUIREMENTS/DESIGN DOCUMENT, 1 July 1985, Contract: DAAK20-81-C-0643, CDRL Item 2-15/004, Volume I, Fifth Submittal (Final) (Revision G), USAMS Algorithm Analysis Document Number ALGO_IGR_0008, page I-206.

Improved GUARDRAIL V MC68000 'DF' Files

2 FILES WITH EXTENSION '.BAT'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFALG.BAT_NUM

2.1 DFALG.BAT_NUM

```
***** Source Listing -> DFALG.BAT_NUM *****
*****
1  SIGNON DFALG,188,DPUCMIGR
2  SE GR 115
3  XDEL DFALG.CNT
4  XDEL DFALG.XXX
5  IGRMIDS DFALG.CMD,,SY3:DFALG.LST
6  L TEXT32,20
7  ST ,COM=DFALG.CLN,LIST=NULL:
8  L TEXT32,20
9  ST ,COM=DFALG.MID,LIST=NULL:
10 L .BG,BASIC/S,100
11 T .BG
12 AS 0,DFALG.CN1
13 ST
14 XDEL DFALG.CMD
15 XDEL DFALG.CLN
16 XDEL DFALG.FIL
17 XDEL DFALG.MID
18 XDEL DFALG.IMB
19 XDEL DFALG.CN1
20 XDEL DFALG.CN2
21 XDEL SY3:DFALG.XXX
22 XDEL SY3:DFALG.LST
23 XDEL SY3:DFALG.TXT
24 SIGNOFF
```


Improved GUARDRAIL V MC68000 'DF' Files

3 FILES WITH EXTENSION '.CMD'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFALG.CMD_NUM

3.1 DFALG.CMD_NUM

```
***** Source Listing --> DFALG.CMD_NUM *****  
*****  
1 INT DFALG.MST  
2 FMT ALL,SY3:DFALG.TXT ,DELETE  
3 ALL  
4 HIR  
5 END
```

Improved GUARDRAIL V MC68000 'DF' Files

4 FILES WITH EXTENSION '.CNT'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFALG.CNT_NUM

4.1 DFALG.CNT_NUM

***** Source Listing —> DFALG.CNT_NUM *****

1	.SKIP		
2	2.3.2	DFALG MASTER FILE	.4205
3	.SKIP		
4	2.3.2.1	ACCBUF.MOD	.4217
5	.SKIP		
6	2.3.2.2	ACCUM.MOD	.4218
7	.SKIP		
8	2.3.2.3	ACUINIT.MOD	.4219
9	.SKIP		
10	2.3.2.4	ACULOAD.MOD	.4220
11	.SKIP		
12	2.3.2.5	ACUSETUP.MOD	.4221
13	.SKIP		
14	2.3.2.6	ADFINIT.MOD	.4222
15	.SKIP		
16	2.3.2.7	ADJFDFC.MOD	.4223
17	.SKIP		
18	2.3.2.8	ARCTAN.MOD	.4224
19	.SKIP		
20	2.3.2.9	BLSGPRS.MOD	.4225
21	.SKIP		
22	2.3.2.10	CALFRQS.MOD	.4227
23	.SKIP		
24	2.3.2.11	CLRACCS.MOD	.4228
25	.SKIP		
26	2.3.2.12	COARSE.MOD	.4229
27	.SKIP		
28	2.3.2.13	CODF.MOD	.4230
29	.SKIP		
30	2.3.2.14	DETBAND.MOD	.4231
31	.SKIP		
32	2.3.2.15	DETSIG.MOD	.4232
33	.SKIP		
34	2.3.2.16	DFA1.MOD	.4233
35	.SKIP		
36	2.3.2.17	DFA2.MOD	.4234
37	.SKIP		
38	2.3.2.18	DFA4.MOD	.4236
39	.SKIP		
40	2.3.2.19	DFA5.MOD	.4237
41	.SKIP		
42	2.3.2.20	DFA6.MOD	.4238
43	.SKIP		

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFALG.CNT_NUM

44	2.3.2.21	DFIN.MOD	.4240
45	.SKIP		
46	2.3.2.22	DFOUT.MOD	.4241
47	.SKIP		
48	2.3.2.23	DFPON.MOD	.4243
49	.SKIP		
50	2.3.2.24	DIRREV.MOD	.4244
51	.SKIP		
52	2.3.2.25	DMPFIFO.MOD	.4245
53	.SKIP		
54	2.3.2.26	DQUEUE.MOD	.4246
55	.SKIP		
56	2.3.2.27	FMTDFR.MOD	.4247
57	.SKIP		
58	2.3.2.28	INITDF.MOD	.4248
59	.SKIP		
60	2.3.2.29	INPQUAD.MOD	.4249
61	.SKIP		
62	2.3.2.30	MULSWLW.MOD	.4250
63	.SKIP		
64	2.3.2.31	NQUEUE.MOD	.4251
65	.SKIP		
66	2.3.2.32	P2SRCH.MOD	.4252
67	.SKIP		
68	2.3.2.33	PARFIT.MOD	.4254
69	.SKIP		
70	2.3.2.34	RDFIFOS.MOD	.4255
71	.SKIP		
72	2.3.2.35	RDFIFOU.MOD	.4256
73	.SKIP		
74	2.3.2.36	READAGC.MOD	.4257
75	.SKIP		
76	2.3.2.37	RFPLOAD.MOD	.4258
77	.SKIP		
78	2.3.2.38	RFPSETUP.MOD	.4259
79	.SKIP		
80	2.3.2.39	SIGPRES.MOD	.4260
81	.SKIP		
82	2.3.2.40	SIOLOAD.MOD	.4262
83	.SKIP		
84	2.3.2.41	SIOSEND.MOD	.4264
85	.SKIP		
86	2.3.2.42	SQTHRESH.MOD	.4265
87	.SKIP		
88	2.3.2.43	STARTDF.MOD	.4266
89	.SKIP		
90	2.3.2.44	STRTDF.MOD	.4267

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFALG.CNT_NUM

91	.SKIP		
92	2.3.2.45	WAITDF.MOD	.4268

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFGRND.CNT_NUM

4.2 DFGRND.CNT_NUM

***** Source Listing —> DFGRND.CNT_NUM *****

1	.SKIP	
2	2.2.2	DFGRND MASTER FILE 3909
3	.SKIP	
4	2.2.2.1	DFG1.MOD 3916
5	.SKIP	
6	2.2.2.2	DFG2.MOD 3918
7	.SKIP	
8	2.2.2.3	DFG3.MOD 3919
9	.SKIP	
10	2.2.2.4	DFG5.MOD 3921
11	.SKIP	
12	2.2.2.5	DFG5T.MOD 3922
13	.SKIP	
14	2.2.2.6	DFG6.MOD 3923
15	.SKIP	
16	2.2.2.7	DFG7.MOD 3925
17	.SKIP	
18	2.2.2.8	DFG8.MOD 3926
19	.SKIP	
20	2.2.2.9	DFG9.MOD 3928
21	.SKIP	
22	2.2.2.10	DFPE1.MOD 3929
23	.SKIP	
24	2.2.2.11	DFPE2.MOD 3930
25	.SKIP	
26	2.2.2.12	FINDOLD.MOD 3932
27	.SKIP	
28	2.2.2.13	GETCRS.MOD 3933
29	.SKIP	
30	2.2.2.14	GETDF.MOD 3934
31	.SKIP	
32	2.2.2.15	GETFINE.MOD 3935
33	.SKIP	
34	2.2.2.16	GETNUM.MOD 3936
35	.SKIP	
36	2.2.2.17	GETPTR.MOD 3937
37	.SKIP	
38	2.2.2.18	PUSH.MOD 3938
39	.SKIP	
40	2.2.2.19	REMBOT.MOD 3939

Improved GUARDRAIL V MC68000 'DF' Files

5 FILES WITH EXTENSION '.CSS'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFOUT.CSS_NUM

5.1 DFOUT.CSS_NUM

```
***** Source Listing —> DFOUT.CSS_NUM *****  
*****  
1 $DEFINE 1,,CURRENT(EOT)  
2 $IFNE 0  
3 ME MSM ----- @*1 ERROR(S) IN DFOUT -----  
4 $ELSE  
5 ME MSM DFOUT DONE. NO ERRORS.  
6 $ENDC  
7 $EXIT
```

Improved GUARDRAIL V MC68000 'DF' Files

6 FILES WITH EXTENSION '.D68'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFBANDS.D68_NUM

6.1 DFBANDS.D68_NUM

```
***** Source Listing -> DFBANDS.D68_NUM *****
*****
1      TTL      'DFBANDS'
2      ; These are the DF bands and maximum roll angle definitions.
3      ; They reside in common so that both the initialization and DF
4      ; software can access them.
5      SECT      ACCDS
6  DFBANDS  IDNT
7      OPT      -M
8      ;
9      XDEF      MAXROLL      ;maximum roll angle
10     XDEF      VHFHI        ;VHF band limit
11     XDEF      UHF          ;UHF band limit
12     ;
13  MAXROLL  DC.W      0      ;maximum roll angle
14  VHFHI    DC.L      0      ;VHF band limit
15  UHF      DC.L      0      ;UHF band limit
16     ;
17     END
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOL.D68_NUM

6.2 DFCOL.D68_NUM

***** Source Listing —> DFCOL.D68_NUM *****

```

1          TTL      'DFCOL'
2      *
3      * DF DATA COLLECTION SEQUENCES FOR COARSE DF ( ALL BANDS )
4      * AND TUNING SEQUENCES FOR ALL BANDS (AND FINE DF ).
5      *
6      DFCOL      IDNT
7      *
8          XDEF      TUNSEQ, TUNEACU, TUNEFDFC
9          XDEF      VLOCOAR, VLOCOAA, VLOCOAF
10         XDEF      VHICOAR, VHICOOA, VHICOAF
11         XDEF      UHFCOAR, UHFCOAA, UHFCOAF
12      *
13      BL1      EQU      1
14      BL2      EQU      2
15      BL3      EQU      3
16      BL4      EQU      4
17      BL5      EQU      5
18      BL6      EQU      6
19      BL7      EQU      7
20      BL8      EQU      8
21      BLE      EQU      9
22      REV      EQU      $80
23      DC      EQU      0          DON'T CARE
24      BLA      EQU      0
25      *
26          SECT      ADFUC
27      *
28      TUNSEQ    DC.W      5          VHF LO
29              DC.B      $1C, DC, $1C, DC, $10, BLA, $90, BLA, $25, BLE
30              DC.W      5          VHF HI
31              DC.B      $1C, DC, $1C, DC, $10, BLA, $90, BLA, $25, BLE
32              DC.W      5          UHF
33              DC.B      $1C, DC, $1C, DC, $0F, BLA, $8F, BLA, $25, BLE
34      *
35      * ACU COMMANDS FOR TUNING SEQUENCE
36      *
37      TUNEACU    DC.W      1          VHF LO
38              DC.W      $0000
39              DC.W      1          VHF HI
40              DC.W      $0000
41              DC.W      0          UHF
42      *
43      * FDFC COMMANDS FOR TUNING SEQUENCE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFCOL.D68_NUM

```

44 *
45 TUNEFDFC DC.W $157D VHF LO
46 DC.W 45<<1
47 DC.W $157D VHF HI
48 DC.W 45<<1
49 DC.W $057D UHF
50 DC.W 45<<1
51 *
52 * VHF LO COARSE DF
53 *
54 VLOCOAR DC.W 15 1ST STEP
55 DC.B $13,BL1+REV,$13,BL1,$01,BL2+REV
56 DC.B $01,BL2,$88,BL3+REV,$08,BL3
57 DC.B $97,BL4+REV,$17,BL4,$8B,BL5+REV
58 DC.B $0B,BL5,$25,BLE,$13,BL1
59 DC.B $13,BL1+REV,$01,BL2,$01,BL2+REV
60 DC.W 8 2ND STEP
61 DC.B $08,BL3,$88,BL3+REV,$17,BL4
62 DC.B $97,BL4+REV,$0B,BL5,$8B,BL5+REV
63 DC.B $25,BLE,$13,BL1+REV
64 DC.W 10 3RD STEP
65 DC.B $13,BL1,$01,BL2+REV,$01,BL2
66 DC.B $88,BL3+REV,$08,BL3,$97,BL4+REV
67 DC.B $17,BL4,$8B,BL5+REV,$0B,BL5
68 DC.B $25,BLE
69 *
70 * ACU COMMANDS FOR VHF LO COARSE
71 *
72 VLOCOAA DC.W 15 1ST STEP
73 DC.W $8610,$8200 1-,1+
74 DC.W $8610,$8200 2-,2+
75 DC.W $8200,$8200 3-,3+
76 DC.W $8200,$8200 4-,4+
77 DC.W $8200,$8200 5-,5+
78 DC.W $8200,$8200 E,1+
79 DC.W $8610,$8200 1-,2+
80 DC.W $8610 2-
81 DC.W 8 2ND STEP
82 DC.W $8200,$8200 3+,3-
83 DC.W $8200,$8200 4+,4-
84 DC.W $8200,$8200 5+,5-
85 DC.W $8200,$8610 E,1-
86 DC.W 3 3RD STEP
87 DC.W $8200,$8610 1+,2-
88 DC.W $8200 2+
89 *
90 * FDFC COMMANDS FOR VHF LO COARSE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFCOL.D68_NUM

```

91 *
92 VLOCOAF DC.W $FF31 1ST STEP
93 DC.W 43<<1
94 DC.W $8821 2ND STEP
95 DC.W 43<<1
96 DC.W $3A21 3RD STEP
97 DC.W 43<<1
98 *
99 * VHF HI COARSE DF
100 *
101 VHICOAR DC.W 15 1ST STEP
102 DC.B $13,BL1,$13,BL1+REV,$13,BL2
103 DC.B $13,BL2+REV,$01,BL3,$01,BL3+REV
104 DC.B $01,BL4,$01,BL4+REV,$13,BL5
105 DC.B $13,BL5+REV,$08,BL6,$88,BL6+REV
106 DC.B $17,BL7,$97,BL7+REV,$0B,BL8
107 DC.W 8 2ND STEP
108 DC.B $8B,BL8+REV,$25,BLE,$13,BL1+REV
109 DC.B $13,BL1,$13,BL2+REV,$13,BL2
110 DC.B $01,BL3+REV,$01,BL3
111 DC.W 11 3RD STEP
112 DC.B $01,BL4+REV,$01,BL4,$13,BL5+REV
113 DC.B $13,BL5,$88,BL6+REV,$08,BL6
114 DC.B $97,BL7+REV,$17,BL7,$8B,BL8+REV
115 DC.B $0B,BL8,$25,BLE
116 *
117 * VHF HI COARSE ACU COMMANDS
118 *
119 VHICOAA DC.W 15 1ST STEP
120 DC.W $5140,$5550,$2CB0 1+,1-,2+
121 DC.W $28A0,$5140,$5550 2-,3+,3-
122 DC.W $2CB0,$28A0,$4D30 4+,4-,5+
123 DC.W $4920,$4100,$4100 5-,6+,6-
124 DC.W $4100,$4100,$4100 7+,7-,8+
125 DC.W 8 2ND STEP
126 DC.W $4100,$8200,$5550 8-,E,1-
127 DC.W $5140,$28A0,$2CB0 1+,2-,2+
128 DC.W $5550,$5140 3-,3+
129 DC.W 11 3RD STEP
130 DC.W $28A0,$2CB0,$4920 4-,4+,5-
131 DC.W $4D30,$4100,$4100 5+,6-,6+
132 DC.W $4100,$4100,$4100 7-,7+,8-
133 DC.W $4100,$8200 8+,E
134 *
135 * VHF HI COARSE FDFC COMMANDS
136 *
137 VHICOAF DC.W $FF31 1ST STEP

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOL.D68_NUM

```

138          DC.W      42<<1
139          DC.W      $8821          2ND STEP
140          DC.W      42<<1
141          DC.W      $BB21          3RD STEP
142          DC.W      41<<1
143      *
144      * UHF COARSE DF
145      *
146      UHFCOAR  DC.W      15          1ST STEP
147          DC.B      $84,BL5+REV,$04,BL5,$97,BL6+REV
148          DC.B      $17,BL6,$8F,BL1+REV,$0F,BL1
149          DC.B      $88,BL2+REV,$08,BL2,$89,BL3+REV
150          DC.B      $09,BL3,$93,BL4+REV,$13,BL4
151          DC.B      $84,BL5+REV,$04,BL5,$97,BL6+REV
152          DC.W      8          2ND STEP
153          DC.B      $17,BL6,$25,BLE,$0F,BL1
154          DC.B      $8F,BL1+REV,$08,BL2,$88,BL2+REV
155          DC.B      $09,BL3,$89,BL3+REV
156          DC.W      11          3RD STEP
157          DC.B      $13,BL4,$93,BL4+REV,$04,BL5
158          DC.B      $84,BL5+REV,$17,BL6,$97,BL6+REV
159          DC.B      $04,BL5,$84,BL5+REV,$17,BL6
160          DC.B      $97,BL6+REV,$25,BLE
161      *
162      * UHF COARSE ACU COMMANDS
163      *
164      UHFCOAA  DC.W      0          1ST STEP
165          DC.W      0          2ND STEP
166          DC.W      0          3RD STEP
167      *
168      * UHF COARSE FDFC COMMANDS
169      *
170      UHFCOAF  DC.W      $0F31          1ST STEP
171          DC.W      42<<1
172          DC.W      $0821          2ND STEP
173          DC.W      42<<1
174          DC.W      $0B21          3RD STEP
175          DC.W      41<<1
176      *
177          END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFCOL2.D68_NUM

6.3 DFCOL2.D68_NUM

***** Source Listing —> DFCOL2.D68_NUM *****

```

1      TTL 'DFCOL2'
2      *
3      * DF DATA COLLECTION SEQUENCES FOR COARSE DF ( ALL BANDS )
4      * AND TUNING SEQUENCES FOR ALL BANDS (AND FINE DF ).
5      *
6      * DFCOL2 - ADJUSTED DFCOL TO DO A COARSE VERIFY
7      *
8      DFCOL2    IDNT
9      *
10     XDEF      VLOCOAR2,VLOCOAA2,VLOCOAF2
11     XDEF      VHICOAR2,VHICOAA2,VHICOAF2
12     XDEF      UHFCOAR2,UHFCOAA2,UHFCOAF2
13     *
14     BL1      EQU      1
15     BL2      EQU      2
16     BL3      EQU      3
17     BL4      EQU      4
18     BL5      EQU      5
19     BL6      EQU      6
20     BL7      EQU      7
21     BL8      EQU      8
22     BLE      EQU      9
23     REV      EQU      $80
24     DC        EQU      0                      DON'T CARE
25     BLA      EQU      0
26     *
27     SECT      ADFUC
28     *
29     *
30     * VHF LO COARSE DF
31     *
32     VLOCOAR2 DC.W    0                      NULL STEP
33     DC.W      12                      1ST STEP
34     DC.B      $13,BL1+REV,$13,BL1,$01,BL2+REV
35     DC.B      $01,BL2,$88,BL3+REV,$08,BL3
36     DC.B      $97,BL4+REV,$17,BL4,$8B,BL5+REV
37     DC.B      $0B,BL5,$25,BLE,$13,BL1
38     DC.W      11                      2ND STEP
39     DC.B      $13,BL1+REV,$01,BL2,$01,BL2+REV
40     DC.B      $08,BL3,$88,BL3+REV,$17,BL4
41     DC.B      $97,BL4+REV,$0B,BL5,$8B,BL5+REV
42     DC.B      $25,BLE,$13,BL1+REV
43     DC.W      10                      3RD STEP

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFCOL2.D68_NUM

```

44      DC.B      $13,BL1,$01,BL2+REV,$01,BL2
45      DC.B      $38,BL3+REV,$08,BL3,$97,BL4+REV
46      DC.B      $17,BL4,$8B,BL5+REV,$0B,BL5
47      DC.B      $25,BLE
48      *
49      * ACU COMMANDS FOR VHF LO COARSE
50      *
51      VLOCOAA2 DC.W      0          NULL STEP
52      DC.W      4          1ST STEP
53      DC.W      $8610,$8200      1-,1+
54      DC.W      $8610,$8200      2-,2+
55      DC.W      11          2ND STEP
56      DC.W      $8610,$8200      1-,2+
57      DC.W      $8610,$8200      2-,3+
58      DC.W      $8200,$8200      3-,4+
59      DC.W      $8200,$8200      4-,5+
60      DC.W      $8200,$8200      5-,E
61      DC.W      $8610          1-
62      DC.W      3          3RD STEP
63      DC.W      $8200,$8610      1+,2-
64      DC.W      $8200          2+
65      *
66      * EDFC COMMANDS FOR VHF LO COARSE
67      *
68      VLOCOAF2 DC.W      $0000      NULL STEP
69      DC.W      0
70      DC.W      $4C31          1ST STEP
71      DC.W      43<<1
72      DC.W      $BB21          2ND STEP
73      DC.W      43<<1
74      DC.W      $3A21          3RD STEP
75      DC.W      43<<1
76      *
77      * VHF HI COARSE DF
78      *
79      VHICOAR2 DC.W      0          NULL STEP
80      DC.W      12          1ST STEP
81      DC.B      $13,BL1,$13,BL1+REV,$13,BL2
82      DC.B      $13,BL2+REV,$01,BL3,$01,BL3+REV
83      DC.B      $01,BL4,$01,BL4+REV,$13,BL5
84      DC.B      $13,BL5+REV,$08,BL6,$88,BL6+REV
85      DC.W      11          2ND STEP
86      DC.B      $17,BL7,$97,BL7+REV,$0B,BL8
87      DC.B      $8B,BL8+REV,$25,BLE,$13,BL1+REV
88      DC.B      $13,BL1,$13,BL2+REV,$13,BL2
89      DC.B      $01,BL3+REV,$01,BL3
90      DC.W      11          3RD STEP

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOL2.D68_NUM

```

91          DC.B      $01,BL4+REV,$01,BL4,$13,BL5+REV
92          DC.B      $13,BL5,$88,BL6+REV,$08,BL6
93          DC.B      $97,BL7+REV,$17,BL7,$8B,BL8+REV
94          DC.B      $0B,BL8,$25,BLE
95      *
96      * VHF HI COARSE ACU COMMANDS
97      *
98      VHICOA2 DC.W      0              NULL STEP
99          DC.W      12              1ST STEP
100         DC.W      $5140,$5550,$2CB0 1+,1-,2+
101         DC.W      $28A0,$5140,$5550 2-,3+,3-
102         DC.W      $2CB0,$28A0,$4D30 4+,4-,5+
103         DC.W      $4920,$4100,$4100 5-,6+,6-
104         DC.W      11              2ND STEP
105         DC.W      $4100,$4100,$4100 7+,7-,8+
106         DC.W      $4100,$8200,$5550 8-,E,1-
107         DC.W      $5140,$28A0,$2CB0 1+,2-,2+
108         DC.W      $5550,$5140      3-,3+
109         DC.W      11              3RD STEP
110         DC.W      $28A0,$2CB0,$4920 4-,4+,5-
111         DC.W      $4D30,$4100,$4100 5+,6-,6+
112         DC.W      $4100,$4100,$4100 7-,7+,8-
113         DC.W      $4100,$8200      8+,E
114      *
115      * VHF HI COARSE FDFC COMMANDS
116      *
117      VHICOF2 DC.W      $0000          NULL STEP
118          DC.W      0
119          DC.W      $CC31          1ST STEP
120          DC.W      42<<1
121          DC.W      $BB21          2ND STEP
122          DC.W      42<<1
123          DC.W      $BB21          3RD STEP
124          DC.W      41<<1
125      *
126      * UHF COARSE DF
127      *
128      UHFCOAR2 DC.W      0              NULL STEP
129          DC.W      12              1ST STEP
130          DC.B      $84,BL5+REV,$04,BL5,$97,BL6+REV
131          DC.B      $17,BL6,$8F,BL1+REV,$0F,BL1
132          DC.B      $88,BL2+REV,$08,BL2,$89,BL3+REV
133          DC.B      $09,BL3,$93,BL4+REV,$13,BL4
134          DC.W      11              2ND STEP
135          DC.B      $84,BL5+REV,$04,BL5,$97,BL6+REV
136          DC.B      $17,BL6,$25,BLE,$0F,BL1
137          DC.B      $8F,BL1+REV,$08,BL2,$88,BL2+REV

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOL2.D68_NUM

```

138      DC.B      $09,BL3,$89,BL3+REV
139      DC.W      11              3RD STEP
140      DC.B      $13,BL4,$93,BL4+REV,$04,BL5
141      DC.B      $84,BL5+REV,$17,BL6,$97,BL6+REV
142      DC.B      $04,BL5,$84,BL5+REV,$17,BL6
143      DC.B      $97,BL6+REV,$25,BLE
144      *
145      * UHF COARSE ACU COMMANDS
146      *
147      UHFCOAA2 DC.W      0              NULL STEP
148      DC.W      0              1ST STEP
149      DC.W      0              2ND STEP
150      DC.W      0              3RD STEP
151      *
152      * UHF COARSE FDFC COMMANDS
153      *
154      UHFCOAF2 DC.W      $0000          NULL STEP
155      DC.W      0
156      DC.W      $0C31          1ST STEP
157      DC.W      42<<1
158      DC.W      $0B21          2ND STEP
159      DC.W      42<<1
160      DC.W      $0B21          3RD STEP
161      DC.W      41<<1
162      *
163      END

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOM.D68_NUM

6.4 DFCOM.D68_NUM

***** Source Listing —> DFCOM.D68_NUM *****

```
1      TTL      'DFCOM'
2      *
3      * THIS DATA BASE IS USED PRIMARILY TO PASS THE
4      * CURRENT NAV DATA FROM THE AADMIN CPU TO THE DF CPU.
5      *
6      DFCOM      IDNT
7                  SECT      ACCDS
8      *
9                  XDEF      NAVDATA
10                 XDEF      CALIB
11                 XDEF      DFRPEN
12      *
13      NAVDATA    DS.L      3      CURRENT NAVDATA
14      CALIB      DS.W      1      CALIB SEG. 1 FLAG
15      DFRPEN     DS.W      1      DF REPORT ENABLE FLAG
16      *
17                  END
```

Improved GUARDRAIL V MC68000 'DF' Files

7 FILES WITH EXTENSION '.DBS'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DF1.DBS_NUM

7.1 DF1.DBS_NUM

***** Source Listing --> DF1.DBS_NUM *****

```

1  *
2  *      DF1      DF REQUEST QUEUE ITEM
3  *
4  *      THERE WILL BE ONE QUEUE FOR EACH OF THE TYPES OF FINE
5  *      DF AND CALIBRATION, AND ONE QUEUE FOR THE COARSE DF
6  *      REQUESTS. THE DIFFERENT QUEUES AND THE CORRESPONDING
7  *      QUEUE DEPTH ARE SHOWN IN THE TABLE. THE QUEUES WILL BE
8  *      IMPLEMENTED AS A SET SIZE (THE DEPTH) OF POINTERS TO
9  *      THE QUEUE ITEMS. THE QUEUES WILL BE IN LOCAL COMMON, AND
10 *      POINTERS TO THE TOP AND BOTTOM OF EACH QUEUE WILL BE
11 *      MAINTAINED.
12 *
13 *      QUEUE      DEPTH
14 *      -----
15 *      BITE      9
16 *      MANUAL    14
17 *      INTEROP.   7
18 *      AUTO DF   14
19 *      G.S.      4
20 *      CALIB.    5
21 *      COARSE    18
22 *
23 *
24 *      STRC
25 *      USES      2,DF1_TIM      TIME ENTERED QUEUE
26 *      USES      1,DF1_MID      MESSAGE ID
27 *      USES      1              UNUSED
28 *      USES      2,DF1_ACC      ACCOUNTABILITY
29 *      USES      1,DF1_TYP      DF TYPE
30 *      USES      1,DF1_ASUB     DF SUB TYPE & AUDIO CORR.
31 *      BITS      1,DF1_AUD     AUDIO CORRELATION TUNE
32 *      BITS      7,DF1_SUB     DF SUB TYPE
33 *      LABEL      DF1_FRQ      FREQUENCY (10 BCD DIGITS)
34 *      USES      1,DF1_FR1     1ST 2 BCD DIGITS
35 *      USES      1,DF1_FR2     2ND 2 BCD DIGITS
36 *      USES      1,DF1_FR3     3RD 2 BCD DIGITS
37 *      USES      1,DF1_FR4     4TH 2 BCD DIGITS
38 *      USES      1,DF1_FR5     LAST 2 BCD DIGITS
39 *      LABEL      DF1_RCVR     RECEIVER R-924 CONTROL
40 *      USES      1,DF1_GAIN     GAIN TYPE & MANUAL CNTRL
41 *      BITS      2,DF1_TYPE     GAIN TYPE
42 *      BITS      6,DF1_MAN     MANUAL GAIN SETTING
43 *      USES      1,DF1_RNG     BANDWIDTH AND DETECTORS

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DF1.DBS_NUM

44	USES	1,DF1_FLAG	MISC FLAGS
45	USES	1,DF1_POR	DS PORTION NO.
46	USES	1,DF1_DSE	DS ENTRY NO.
47	USES	4,DF1_SIG	GS SIGNAL TYPE BIT MAP
48	USES	1,DF1_ARF	ARF ID.
49	USES	1,DF1_BND	DS BANDWIDTH
50	USES	1,DF1_ADFC	AUTO DF COUNTER
51	USES	1,DF1_ARTF	ART FLAG
52	ENDST	DF1	

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFC.DBS_NUM

7.2 DFC.DBS_NUM

***** Source Listing —> DFC.DBS_NUM *****

```
1  *
2  *      DFC    DF DATA COLLECTION SEQUENCE HEADER.
3  *
4  *      EACH HEADER BLOCK CONTAINS THE POINTERS TO
5  *      THE APPROPRIATE COMMAND SEQUENCE FOR THE
6  *      RF PROCESSOR, ACU'S, AND FAST DF CONTROLLER.
7  *
8
9      STRC
10     USES      2,DFC_NUM      NUMBER OF STEPS IN SEQUENCE
11     USES      4,DFC_RFP     PTR TO RF PROCESSOR COMMANDS
12     USES      4,DFC_ACU     PTR TO ACU COMMANDS
13     USES      4,DFC_FDFC    PTR TO FDFC COMMANDS
14     ENDST      DFC
```


Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFQ.DBS_NUM

7.3 DFQ.DBS_NUM

***** Source Listing —> DFQ.DBS_NUM *****

```
1  *
2  *      DFQ    DF DATA COLLECTION QUEUE
3  *
4  *      THIS QUEUE IS 2 ELEMENTS DEEP AND
5  *      CONTAINS THE PARAMETERS TO IDENTIFY
6  *      EACH DF DATA COLLECTION STEP IN PROCESS.
7  *
8  *      STRC
9  *      USES    4,DFQ_PTR      POINTER TO RF PROC. COMMANDS
10 *      USES    2,DFQ_STP      STEP NUMBER OF DF STEP
11 *      USES    2,DFQ_FDF      ACCUM. MODE (A&C,C ONLY)
12 *      ENDST    DFQ
```

Improved GUARDRAIL V MC68000 'DF' Files

8 FILES WITH EXTENSION '.DOC'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFALG.DOC_NUM

8.1 DFALG.DOC_NUM

***** Source Listing —> DFALG.DOC_NUM *****

```
1 2.3.2 DFALG MASTER FILE
2 .SK 2
3 This is the software maintenance documentation for the DFALG
4 Master File.
5 .SK 2
6 The major jobs involved in the DF algorithm are:
7 starting the data collection, cycling through
8 the data collection, accumulating the data in the
9 case of Fine or Calibration DF requests, calculating
10 the LOB, and transmitting the data down to the GDPU.
11 .SK2
12 This section contains
13 a MIDS module catalog listing of the modules in DFALG,
14 and a MIDS module description for each module within
15 DFALG.
16 .EJ
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFGRND.DOC_NUM

8.2 DFGRND.DOC_NUM

***** Source Listing --> DFGRND.DOC_NUM *****

```
1 2.2.2 DFGRND MASTER FILE
2 .SK 2
3 This is the software maintenance documentation for the DFGRND
4 Master File.
5 .SK 2
6 The major jobs involved in the DF scheduling
7 are: DF requests from the Perkin
8 Elmer, DF requests from within the GDPU, scheduling
9 DF requests, removing requests that have not been
10 processed in a set amount of time, controlling
11 the signal generator (for calibration) and controlling
12 the audio correlator, and reporting the DF response
13 to the Perkin Elmer or the appropriate job.
14 .SK2
15 This section contains
16 a MIDS module catalog listing of the modules in DFGRND,
17 and a MIDS module description for each module within
18 DFGRND.
19 .EJ
```

Improved GUARDRAIL V MC68000 'DF' Files

9 FILES WITH EXTENSION '.EQU'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFERRS.EQU_NUM

9.1 DFERRS.EQU_NUM

```
***** Source Listing --> DFERRS.EQU_NUM *****
*****
1  *
2  * DF ERRORS BIT ASSIGNMENTS
3  *
4  DFREJ    EQU    0          DF REJECTED (PREV. TOO LONG )
5  TOOLNG    EQU    1          DF DID NOT COMPLETE IN SEGMENT
6  NOSIGNL   EQU    2          NO SIGNAL PRESENT
7  BLSGF     EQU    3          BASE LINE SIG PRES FAILED
8  BLSGA     EQU    4          BASE LINE SIG PRES ABORT
9  ROLLANG   EQU    5          ROLL ANGLE EXCEEDS LIMIT
10 MULT      EQU    6          MULTIPLE MINIMA IN 1ST SEARCH
11 PFIT      EQU    7          POOR FIT, BAD QUALITY
12 MISSEG    EQU    8          MISSING SEGMENT (FINE DF)
13 NOCAL     EQU    9          NO CAL TABLES
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFTYPE.EQU_NUM

9.2 DFTYPE.EQU_NUM

```
***** Source Listing --> DFTYPE.EQU_NUM *****
*****
1  *
2  * DF TYPE CODES
3  *
4  TBITE      EQU      1
5  TMANUAL    EQU      2
6  TINTEROP   EQU      3
7  TAUTO      EQU      4
8  TGS         EQU      5
9  TCALIB     EQU      6
10 TAOI        EQU      7
11 TGEO        EQU      8
```

Improved GUARDRAIL V MC68000 'DF' Files

10 FILES WITH EXTENSION '.INC'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFERR.INC_NUM

10.1 DFERR.INC_NUM

***** Source Listing --> DFERR.INC_NUM *****

1	*	DF ERRORS BIT MAP BIT ASSIGNMENTS
2	*	
3	*	0 DF REJECTED (PREV. TOO LONG)
4	*	1 DF DID NOT COMPLETE IN SEGMENT
5	*	2 NO SIGNAL PRESENT
6	*	3 BASE LINE SIG PRES FAILED
7	*	4 BASE LINE SIG PRES ABORT
8	*	5 ROLL ANGLE EXCEEDS LIMIT
9	*	6 MULTIPLE MINIMA IN 1ST SEARCH
10	*	7 POOR FIT, BAD QUALITY
11	*	8 MISSING SEGMENT (FINE DF)
12	*	9 NO CAL TABLES
13	*	

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFTYPE.INC_NUM

10.2 DFTYPE.INC_NUM

***** Source Listing --> DFTYPE.INC_NUM *****

```
1 *   DF TYPE VALUES
2 *
3 *   1 - BITE
4 *   2 - MANUAL DF
5 *   3 - INTEROP
6 *   4 - AUTO DF (DS)
7 *   5 - GENERAL SEARCH (FINE)
8 *   6 - CALIBRATION
9 *   7 - AOI
10 *   8 - GEOSCREEN
11 *
```

Improved GUARDRAIL V MC68000 'DF' Files

11 FILES WITH EXTENSION '.MOD'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA1.MOD_NUM

11.1 DFA1.MOD_NUM

```
***** Source Listing -> DFA1.MOD_NUM *****
*****
1  *O2**MODULE HEADER**
2  *A***HOME MASTER FILE***
3  DFALG.MST
4  *B***MODULE NAME***
5  DFA1.MOD
6  *D***PROGRAMMER***
7  ETO
8  *E***MODULE DESCRIPTOR***
9  DF request input (ARF).
10 *G***NARRATIVE***
11 .imbed sy4:DFA1.STR/G
12 *I***CALLING SEQUENCE***
13 .skip
14 ?SCHED DFA1,#2,,,IPACK
15 .SKIP
16 IPACK=Int*4/Input
17 Pointer to packet containing DF request. Packet
18 definition "DF".
19 *L***FILES USED***
20 *M***EXTERNAL REFERENCES***
21 *W***HIGHER REFERENCES (1/LINE)***
22 *X***LOWER REFERENCES (1/LINE)***
23 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA2.MOD_NUM

11.2 DFA2.MOD_NUM

***** Source Listing —> DFA2.MOD_NUM *****

```
1  *O2**MODULE HEADER**
2  *A***HOME MASTER FILE***
3  DFALG.MST
4  *B***MODULE NAME***
5  DFA2.MOD
6  *D***PROGRAMMER***
7  ETO
8  *E***MODULE DESCRIPTOR***
9  Start/continue DF data collection
10 *G***NARRATIVE***
11 .imbed sy4:DFA2.STR/G
12 *I***CALLING SEQUENCE***
13 .skip
14 ?SCHED DFA2,#1,,
15 .SKIP
16 *L***FILES USED***
17 *M***EXTERNAL REFERENCES***
18 *W***HIGHER REFERENCES (1/LINE)***
19 *X***LOWER REFERENCES (1/LINE)***
20 DETBAND.MOD
21 GETBITS.MOD
22 GETLPAK.MOD
23 PUTPAK.MOD
24 CLRACCS.MOD
25 INITDF.MOD
26 STARTDF.MOD
27 WAITDF.MOD
28 FMTFRQ.MOD
29 SIOLOAD.MOD
30 SIOSEND.MOD
31 RFPSETUP.MOD
32 ACUSETUP.MOD
33 NQUEUE.MOD
34 DQUEUE.MOD
35 ADJFDFC.MOD
36 DMPFIFO.MOD
37 ACCBUF.MOD
38 ACCUM.MOD
39 SIGPRES.MOD
40 READAGC.MOD
41 RDFIFOU.MOD
42 STRTDF.MOD
43 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA4.MOD_NUM

11.3 DFA4.MOD_NUM

***** Source Listing --> DFA4.MOD_NUM *****

```
1  *O2**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFALG.MST
4  *B***MODULE NAME***
5  DFA4.MOD
6  *D***PROGRAMMER***
7  ETO
8  *E***MODULE DESCRIPTOR***
9  Navigation data input.
10 *G***NARRATIVE***
11 .imbed sy4:DFA4.STR/G
12 *I***CALLING SEQUENCE***
13 .skip
14 ?SCHED DFA4,#3,,,
15 .SKIP
16 *L***FILES USED***
17 *M***EXTERNAL REFERENCES***
18 *W***HIGHER REFERENCES (1/LINE)***
19 *X***LOWER REFERENCES (1/LINE)***
20 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA5.MOD_NUM

11.4 DFA5.MOD_NUM

***** Source Listing —> DFA5.MOD_NUM *****

```
1  *02**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFALG.MST
4  *B***MODULE NAME***
5  DFA5.MOD
6  *D***PROGRAMMER***
7  ETO
8  *E***MODULE DESCRIPTOR***
9  Accumulate data for Fine DF.
10 *G***NARRATIVE***
11 .imbed sy4:DFA5.STR/G
12 *I***CALLING SEQUENCE***
13 .skip
14 ?SCHED DFA5,*3,,,IPACK
15 .SKIP
16 IPACK=Int*4/Input
17 Pointer to packet containing baseline data. Packet
18 definition "SD".
19 *L***FILES USED***
20 *M***EXTERNAL REFERENCES***
21 *W***HIGHER REFERENCES (1/LINE)***
22 *X***LOWER REFERENCES (1/LINE)***
23 CPUSEND.MOD
24 GETPAK.MOD
25 INPQUAD.MOD
26 DIRREV.MOD
27 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA6.MOD_NUM

11.5 DFA6.MOD_NUM

***** Source Listing —> DFA6.MOD_NUM *****

```
1  *O2**MODULE HEADER**
2  *A***HOME MASTER FILE**
3  DFALG.MST
4  *B***MODULE NAME**
5  DFA6.MOD
6  *D***PROGRAMMER**
7  ETO
8  *E***MODULE DESCRIPTOR**
9  LOP calculation
10 *G***NARRATIVE**
11 .imbed sy4:DFA6.STR/G
12 *I***CALLING SEQUENCE**
13 .skip
14 ?SCHED   DFA6,#3,,IPACK
15 .SKIP
16 IPACK=Int*4/Input
17 Pointer to packet containing baseline phase
18 angles. Either packet definition "TFA" or
19 "CDF".
20 *L***FILES USED**
21 *M***EXTERNAL REFERENCES**
22 *W***HIGHER REFERENCES (1/LINE)**
23 *X***LOWER REFERENCES (1/LINE)**
24 BLSGPRS.MOD
25 SQTHRESH.MOD
26 BCDBIN.MOD
27 COARSE.MOD
28 FMTDFR.MOD
29 P2SRCH.MOD
30 PARFIT.MOD
31 GETPAK.MOD
32 CPUSEND.MOD
33 *$***END**
```


Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG1.MOD_NUM

11.6 DFG1.MOD_NUM

***** Source Listing --> DFG1.MOD_NUM *****

```
1  *02**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFG1.MST
4  *B***MODULE NAME***
5  DFG1.MOD
6  *D***PROGRAMMER***
7  P. Eto
8  *E***MODULE DESCRIPTOR***
9  DF request input.
10 *G***NARRATIVE***
11 .imbed SY4:DFG1.STR/G
12 *I***CALLING SEQUENCE***
13
14 ?SCHED DFG1,#4,,,IPACK
15 IPACK=Int*4/Input
16 Packet pointer to DF request. Packet definition "DF" or "DG".
17
18 *L***FILES USED***
19 *M***EXTERNAL REFERENCES***
20 *W***HIGHER REFERENCES (1/LINE)***
21 DFPE1.MOD
22 ADFRQST.MOD
23 AOIDF.MOD
24 SCARONE.MOD
25 SCARTWO.MOD
26 ARFDFBIT.MOD
27 GEORQST.MOD
28 GEOCOL.MOD
29 DSGRH.MOD
30 *X***LOWER REFERENCES (1/LINE)***
31 GETNUM.MOD
32 GETPTR.MOD
33 DFG2.MOD
34 REMBOT.MOD
35 ACCOUNT.MOD
36 PUSH.MOD
37 PUTPAK.MOD
38 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG2.MOD_NUM

11.7 DFG2.MOD_NUM

***** Source Listing —> DFG2.MOD_NUM *****

```
1  *O2**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFGRND.MST
4  *B***MODULE NAME***
5  DFG2.MOD
6  *D***PROGRAMMER***
7  P. Eto
8  *E***MODULE DESCRIPTOR***
9  Remove old DF requests.
10 *G***NARRATIVE***
11 .imbed SY4:DFG2.STR/G
12 *I***CALLING SEQUENCE***
13
14 ?SCHED DFG2,#4,,,
15
16 *L***FILES USED***
17 *M***EXTERNAL REFERENCES***
18 HB=Database
19 Header block data base for DF queues.
20 *W***HIGHER REFERENCES (1/LINE)***
21 DFG2.MOD
22 DFG1.MOD
23 *X***LOWER REFERENCES (1/LINE)***
24 DFG2.MOD
25 DFPE2.MOD
26 FINDOLD.MOD
27 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG3.MOD_NUM

11.8 DFG3.MOD_NUM

```
***** Source Listing —> DFG3.MOD_NUM *****
*****
1  *02**MODULE HEADER**
2  *A***HOME MASTER FILE***
3  DFGRND.MST
4  *B***MODULE NAME***
5  DFG3.MOD
6  *D***PROGRAMMER***
7  ETO
8  *E***MODULE DESCRIPTOR***
9  DF scheduler.
10 *G***NARRATIVE***
11 .imbed SY4:DFG3.STR/G
12 *I***CALLING SEQUENCE***
13 .skip
14 JSR DFG3
15 .skip
16 *L***FILES USED***
17 *M***EXTERNAL REFERENCES***
18 *W***HIGHER REFERENCES (1/LINE)***
19 GAQUEUER.MOD
20 DFPE2.MOD
21 *X***LOWER REFERENCES (1/LINE)***
22 GETCRS.MOD
23 GETFINE.MOD
24 GETBITS.MOD
25 PUTBITS.MOD
26 GETPAK.MOD
27 GASEND.MOD
28 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG5.MOD_NUM

11.9 DFG5.MOD_NUM

***** Source Listing —> DFG5.MOD_NUM *****

```
1  *O2**MODULE HEADER**
2  *A***HOME MASTER FILE**
3  DFGRND.MST
4  *B***MODULE NAME**
5  DFG5.MOD
6  *D***PROGRAMMER**
7  SWANSON
8  *E***MODULE DESCRIPTOR**
9  Signal generator control.
10 *G***NARRATIVE**
11 This job controls the signal generator for
12 calibration/verification and BITE DF requests.
13 *I***CALLING SEQUENCE**
14 .skip
15 ?SCHED DFG5,#2,,,IPACK
16 .SKIP
17 IPACK=Int*4/Input.
18 Packet pointer to signal generator request. Packet
19 definition "DFF".
20 *L***FILES USED**
21 *M***EXTERNAL REFERENCES**
22 *W***HIGHER REFERENCES (1/LINE)**
23 DFG6.MOD
24 *X***LOWER REFERENCES (1/LINE)**
25 RELPAK.MOD
26 BCDBIN.MOD
27 DFG5T.MOD
28 *$***END**
```

Improved GUARDRAIL V MC68000 'DF' Files
DRCl:[ALGO.IGR.MC68000_DF]DFG5T.MOD_NUM

11.10 DFG5T.MOD_NUM

```
***** Source Listing --> DFG5T.MOD_NUM *****
*****
1  *O2**MODULE HEADER**
2  *A**HOME MASTER FILE**
3  DFGRND.MST
4  *B**MODULE NAME**
5  DFG5T.MOD
6  *D**PROGRAMMER**
7  SWANSON
8  *E**MODULE DESCRIPTOR**
9  Signal Generator timeout
10 *G**NARRATIVE**
11 This job turns off the Signal Generator if it has
12 not been retuned within the specified timeout length.
13 *I**CALLING SEQUENCE**
14 .sk
15 ?SCHED DFG5T,#4,#100,#1
16 .sk
17 *L**FILES USED**
18 *M**EXTERNAL REFERENCES**
19 NONE
20 *W**HIGHER REFERENCES (1/LINE)**
21 DFG5.MOD
22 DFG5T.MOD
23 *X**LOWER REFERENCES (1/LINE)**
24 DFG5T.MOD
25 *$**END**
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG6.MOD_NUM

11.11 DFG6.MOD_NUM

***** Source Listing —> DFG6.MOD_NUM *****

```
1  *O2**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFGRND.MST
4  *B***MODULE NAME***
5  DFG6.MOD
6  *D***PROGRAMMER***
7  SWANSON
8  *E***MODULE DESCRIPTOR***
9  DF immediate job.
10 *G***NARRATIVE***
11 .imbed SY4:DFG6.STR/G
12 *I***CALLING SEQUENCE***
13 .skip
14 ?SCHED DFG6,#2,,,
15 .SKIP
16 *L***FILES USED***
17 *M***EXTERNAL REFERENCES***
18 SIGREQ=Flag
19 Signal Generator request flag
20 ACORREQ=Flag
21 Audio Correlator request flag.
22 ACORAV=Flag
23 Audio Correlator availability.
24 CURDF=Database
25 Current DF frequency.
26 *W***HIGHER REFERENCES (1/LINE)***
27 GAQUEUER.MOD
28 *X***LOWER REFERENCES (1/LINE)***
29 GETPAK.MOD
30 DFG5.MOD
31 DFG8.MOD
32 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG7.MOD_NUM

11.12 DFG7.MOD_NUM

```
***** Source Listing -> DFG7.MOD_NUM *****
*****
1  *O2**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFGRND.MST
4  *B***MODULE NAME***
5  DFG7.MOD
6  *D***PROGRAMMER***
7  ETO
8  *E***MODULE DESCRIPTOR***
9  Audio correlator timeout.
10 *G***NARRATIVE***
11 .imbed SY4:DFG7.STR/G
12 *I***CALLING SEQUENCE***
13 .skip
14 ?SCHED DFG7,#4,,,
15 .skip
16 *L***FILES USED***
17 *M***EXTERNAL REFERENCES***
18 *W***HIGHER REFERENCES (1/LINE)***
19 *X***LOWER REFERENCES (1/LINE)***
20 GETPAK.MOD
21 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG8.MOD_NUM

11.13 DFG8.MOD_NUM

***** Source Listing —> DFG8.MOD_NUM *****

```
1  *02**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFGRND.MST
4  *B***MODULE NAME***
5  DFG8.MOD
6  *D***PROGRAMMER***
7  M. Swanson
8  *E***MODULE DESCRIPTOR***
9  Audio correlator control.
10 *G***NARRATIVE***
11 .imbed SY4:DFG8.STR/G
12 *I***CALLING SEQUENCE***
13
14 ?SCHED  DFG8,#2,,,IPACK
15
16 IPACK=Int*4/Input
17 Pointer to packet containing Audio Correlator
18 request. Packet definition "ACCR".
19 *L***FILES USED***
20 *M***EXTERNAL REFERENCES***
21 ACORAV=Flag
22 Audio Correlator availability.
23 *W***HIGHER REFERENCES (1/LINE)***
24 ACBITE.MOD
25 DFG6.MOD
26 *X***LOWER REFERENCES (1/LINE)***
27 RELPAK.MOD
28 GETPAK.MOD
29 DFPE2.MOD
30 ACBITR.MOD
31 *$***END***
```


Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG9.MOD_NUM

11.14 DFG9.MOD_NUM

***** Source Listing —> DFG9.MOD_NUM *****

```
1  *O2**MODULE HEADER**
2  *A***HOME MASTER FILE**
3  DFGRND.MST
4  *B***MODULE NAME**
5  DFG9.MOD
6  *D***PROGRAMMER**
7  P. Eto
8  *E***MODULE DESCRIPTOR**
9  Input Audio correlation.
10 *G***NARRATIVE**
11 .imbed SY4:DFG9.STR/G
12 *I***CALLING SEQUENCE**
13 ?SCHED DFG9,#4,,,
14 *L***FILES USED**
15 *M***EXTERNAL REFERENCES**
16 *W***HIGHER REFERENCES (1/LINE)**
17 *X***LOWER REFERENCES (1/LINE)**
18 ACBITR.MOD
19 GETPAK.MOD
20 *$***END**
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFIN.MOD_NUM

11.15 DFIN.MOD_NUM

```
***** Source Listing -> DFIN.MOD_NUM *****
*****
1  *02**MODULE HEADER**
2  *A***HOME MASTER FILE**
3  LIB.MST
4  *B***MODULE NAME**
5  DFIN.MOD
6  *D***PROGRAMMER**
7  P. Wong
8  *E***MODULE DESCRIPTOR**
9  Receives inter-cpu messages on airborne DF CPU.
10 *G***NARRATIVE**
11 This routine receives messages from another CPU.
12 It is executed when the airborne DF CPU is interrupted.
13 The first byte of the global buffer used to pass the message holds
14 the
15 packet size (large or small) of the message.
16 The correct size local packet is allocated
17 and the message is moved into it.
18 The action routine for the incoming packet is retrieved
19 from the TIB table using the offset passed in
20 the global buffer.
21 That routine is then scheduled with the packet.
22 *I***CALLING SEQUENCE**
23 Initiated by interrupt from another CPU.
24 *L***FILES USED**
25 *M***EXTERNAL REFERENCES**
26 ACPUTIB=Database
27 Table of TIB pointers for all routines on
28 the airborne DPU's.
29 BUFD FBSY=Flag
30 Buffer in use indicator.
31 BUFD F=Buffer
32 Global buffer to receive messages.
33 BUFD F DON=Flag
34 Indicates buffer is completely full.
35 This allows detection of spurious interrupts if it is not set.
36 BUFD F OFF=Integer
37 Offset into the TIB pointer table for the desired
38 routine on the ADF CPU.
39 *W***HIGHER REFERENCES (1/LINE)**
40 *X***LOWER REFERENCES (1/LINE)**
41 ADFBIT.MOD
42 PACBUF.MOD
43 *$***END**
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFOUT.MOD_NUM

11.16 DFOUT.MOD_NUM

```
***** Source Listing --> DFOUT.MOD_NUM *****
*****
1  *O2**MODULE HEADER**
2  *A**HOME MASTER FILE**
3  LIB.MST
4  *B**MODULE NAME**
5  DFOUT.MOD
6  *D**PROGRAMMER**
7  WONG
8  *E**MODULE DESCRIPTOR**
9  Send inter-CPU messages.
10 *G**NARRATIVE**
11 This routine sends message packets to the airborne DF CPU.
12 This routine will be initiated by the scheduling of
13 any routine on the ADF CPU from another CPU.
14 The message is written into a global buffer so the local packet can be
15 released. The ADF CPU is then interrupted so it
16 can read the waiting message.
17 *I**CALLING SEQUENCE**
18 .skip
19 ?SCHD ADF CPU routine,#?,,,PACADDR
20 PACADDR=INT*4/INPUT
21 Pointer to any message packet
22 .skip
23 *L**FILES USED**
24 *M**EXTERNAL REFERENCES**
25 BUFD FBSY=Flag
26 Buffer in use indicator.
27 BUFD FWHO=Flag
28 Identifies which CPU is currently using BUFD F.
29 BUFD F=Buffer
30 Global buffer to pass messages to the DF CPU.
31 BUFD FDON=Flag
32 Indicates if buffer is completely full.
33 This allows detection of spurious interrupts.
34 BUFD FOFF=Integer
35 Offset into the TIB pointer table for the desired
36 routine on the ADF CPU.
37 ACPUTIB=Database
38 Table of TIB pointers for all routines the
39 airborne CPUs.
40 *W**HIGHER REFERENCES (1/LINE)**
41 *X**LOWER REFERENCES (1/LINE)**
42 RELPAK.MOD
43 BUFPAC.MOD
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFOUT.MOD_NUM

44 *\$***END***

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFPE1.MOD_NUM

11.17 DFPE1.MOD_NUM

***** Source Listing —> DFPE1.MOD_NUM *****

```
1  *O2**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFGRND.MST
4  *B***MODULE NAME***
5  DFPE1.MOD
6  *D***PROGRAMMER***
7  P. Eto
8  *E***MODULE DESCRIPTOR***
9  Receive DF requests from the Perkin Elmer
10 *G***NARRATIVE***
11 .imbed SY4:DFPE1.STR/G
12 *I***CALLING SEQUENCE***
13
14 ?SCHED DFPE1,#4,,,IPACK
15
16 IPACK=Int*4/Input
17 Packet pointer to DF request. Packet definition "DP".
18
19 *L***FILES USED***
20 *M***EXTERNAL REFERENCES***
21 *W***HIGHER REFERENCES (1/LINE)***
22 *X***LOWER REFERENCES (1/LINE)***
23 GETPAK.MOD
24 DFG1.MOD
25 PUTPAK.MOD
26 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFPE2.MOD_NUM

11.18 DFPE2.MOD_NUM

***** Source Listing —> DFPE2.MOD_NUM *****

```
1  *O2**MODULE HEADER***
2  *A***HOME MASTER FILE***
3  DFGRND.MST
4  *B***MODULE NAME***
5  DFPE2.MOD
6  *D***PROGRAMMER***
7  M. Swanson
8  *E***MODULE DESCRIPTOR***
9  Report DF to Perkin Elmer.
10 *G***NARRATIVE***
11 .imbed SY4:DFPE2.STR/G
12 *I***CALLING SEQUENCE***
13 ?SCHED DFPE2,#4,,IPACK
14 IPACK=Int*4/Input
15 Pointer to packet. Packet could either be
16 1) DF rejected message ("DFRJ"), 2) Audio
17 correlation results ("ACC"), or 3) control
18 packet with DF results ("CTL" and "DFR").
19 *L***FILES USED***
20 *M***EXTERNAL REFERENCES***
21 INTEROP=Flag
22 Interop DF flag received.
23 *W***HIGHER REFERENCES (1/LINE)***
24 GATIM.MOD
25 DFG2.MOD
26 DFG8.MOD
27 REMBOT.MOD
28 *X***LOWER REFERENCES (1/LINE)***
29 GETLPAK.MOD
30 GETPAK.MOD
31 GPOUT.MOD
32 RELPAK.MOD
33 DFG3.MOD
34 *$***END***
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFPON.MOD_NUM

11.19 DFPON.MOD_NUM

```
***** Source Listing —> DFPON.MOD_NUM *****
*****
1  *O2**MODULE HEADER**
2  *A***HOME MASTER FILE**
3  DFALG.MST
4  *B***MODULE NAME**
5  DFPON.MOD
6  *D***PROGRAMMER**
7  ETO
8  *E***MODULE DESCRIPTOR**
9  Power on initialization for DF system.
10 *G***NARRATIVE**
11 This module performs the power on initialization for
12 the DF data collection. The SIO and PIO are set up for
13 use with the Fast DF Controller. A DRCB is set up so
14 that the DF interrupt service routine can perform ARTE
15 calls. The Fast DF Controller is also initialized.
16 *I***CALLING SEQUENCE**
17 .skip
18 JSR DFPON
19 .SKIP
20 *L***FILES USED**
21 *M***EXTERNAL REFERENCES**
22 *W***HIGHER REFERENCES (1/LINE)**
23 *X***LOWER REFERENCES (1/LINE)**
24 INITDF.MOD
25 ACUINIT.MOD
26 *$***END**
```

Improved GUARDRAIL V MC68000 'DF' Files

12 FILES WITH EXTENSION '.MST'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO.IGR.MC68000_DF]DFALG.MST_NUM

12.1 DFALG.MST_NUM

```
***** Source Listing --> DFALG.MST_NUM *****
*****
1  *01**MASTER FILE HEADER***
2  *A***MASTER FILE NAME***
3  DFALG.MST
4  *B***MASTER FILE DESCRIPTOR***
5  *C***% DESIGNED***
6  *D***% CODED***
7  *E***% DEBUGGED***
8  *F***% TESTED***
9  *G***REVISIONS (WHEN, WHY)***
10 *?***TEXT FILE DEFINITION***
11 MODULE NAME
12 *B***MODULE NAME***
13 PROGRAMMER
14 *D***PROGRAMMER***
15 DESCRIPTION
16 *G***NARRATIVE***
17 CALLING SEQUENCE
18 *I***CALLING SEQUENCE***
19 DISC FILES USED
20 *L***FILES USED***
21 EXTERNAL AND COMMON REFERENCES
22 *M***EXTERNAL REFERENCES***
23 HIGHER REFERENCES
24 *W***HIGHER REFERENCES (1/LINE)***
25 LOWER REFERENCES
26 *X***LOWER REFERENCES (1/LINE)***
27 *!***MODULE FORMAT DEFINITION***
28 *02**MODULE HEADER***
29 *A***HOME MASTER FILE***
30 *B***MODULE NAME***
31 *D***PROGRAMMER***
32 *E***MODULE DESCRIPTOR***
33 *G***NARRATIVE***
34 *I***CALLING SEQUENCE***
35 *L***FILES USED***
36 *M***EXTERNAL REFERENCES***
37 *W***HIGHER REFERENCES (1/LINE)***
38 *X***LOWER REFERENCES (1/LINE)***
39 *$***END***
40 *%***ASSOCIATED MODULES (ALPHABETICALLY ORDERED)***
41 8ACCBUF.MOD
42 8ACCUM.MOD
43 8ACUINIT.MOD
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFALG.MST_NUM

44 8ACULOAD.MOD
45 8ACUSETUP.MOD
46 8ADFINIT.MOD
47 8ADJFDFC.MOD
48 8ARCTAN.MOD
49 8BLSGPRS.MOD
50 8CALFRQS.MOD
51 8CLRACCS.MOD
52 8COARSE.MOD
53 8CODF.MOD
54 8DET BAND.MOD
55 8DETSIG.MOD
56 8DFA1.MOD
57 8DFA2.MOD
58 8DFA4.MOD
59 8DFA5.MOD
60 8DFA6.MOD
61 8DFIN.MOD
62 8DFOUT.MOD
63 8DFPON.MOD
64 8DIRREV.MOD
65 8DMPFIFO.MOD
66 8DQUEUE.MOD
67 8FMTDFR.MOD
68 8INITDF.MOD
69 8INPQUAD.MOD
70 8MULSWLW.MOD
71 8NQUEUE.MOD
72 8P2SRCH.MOD
73 8PARFIT.MOD
74 8RDFIFOS.MOD
75 8RDFIFOU.MOD
76 8READAGC.MOD
77 8RFPLOAD.MOD
78 8RFPSETUP.MOD
79 8SIGPRES.MOD
80 8SIOLOAD.MOD
81 8SIOSEND.MOD
82 8SQTHRESH.MOD
83 8STARTDF.MOD
84 8STRTDF.MOD
85 8WAITDF.MOD
86 *\$***END***
87
88
89
90

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFGRND.MST_NUM

12.2 DFGRND.MST_NUM

```
***** Source Listing -> DFGRND.MST_NUM *****
*****
1  *01**MASTER FILE HEADER***
2  *A**MASTER FILE NAME***
3  DFGRND.MST
4  *B**MASTER FILE DESCRIPTOR***
5  *C**% DESIGNED***
6  *D**% CODED***
7  *E**% DEBUGGED***
8  *F**% TESTED***
9  *G**REVISIONS (WHEN, WHY)***
10 *?*TEXT FILE DEFINITION***
11 MODULE NAME
12 *B**MODULE NAME***
13 PROGRAMMER
14 *D**PROGRAMMER***
15 DESCRIPTION
16 *G**NARRATIVE***
17 CALLING SEQUENCE
18 *I**CALLING SEQUENCE***
19 DISC FILES USED
20 *L**FILES USED***
21 EXTERNAL AND COMMON REFERENCES
22 *M**EXTERNAL REFERENCES***
23 HIGHER REFERENCES
24 *W**HIGHER REFERENCES (1/LINE)***
25 LOWER REFERENCES
26 *X**LOWER REFERENCES (1/LINE)***
27 *!**MODULE FORMAT DEFINITION***
28 *02**MODULE HEADER***
29 *A**HOME MASTER FILE***
30 *B**MODULE NAME***
31 *D**PROGRAMMER***
32 *E**MODULE DESCRIPTOR***
33 *G**NARRATIVE***
34 *I**CALLING SEQUENCE***
35 *L**FILES USED***
36 *M**EXTERNAL REFERENCES***
37 *W**HIGHER REFERENCES (1/LINE)***
38 *X**LOWER REFERENCES (1/LINE)***
39 *$**END***
40 *%**ASSOCIATED MODULES (ALPHABETICALLY ORDERED)***
41 8DFG1.MOD
42 8DFG2.MOD
43 8DFG3.MOD
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFGRND.MST_NUM

44 8DFG5.MOD
45 8DFG5T.MOD
46 8DFG6.MOD
47 8DFG7.MOD
48 8DFG8.MOD
49 8DFG9.MOD
50 8DFPE1.MOD
51 8DFPE2.MOD
52 8FINDOLD.MOD
53 8GETCRS.MOD
54 8GETDF.MOD
55 8GETFINE.MOD
56 8GETNUM.MOD
57 8GETPTR.MOD
58 8PUSH.MOD
59 8REMBOT.MOD
60 *\$***END***

Improved GUARDRAIL V MC68000 'DF' Files

13 FILES WITH EXTENSION '.PAC'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DF.PAC_NUM

13.1 DF.PAC_NUM

***** Source Listing ==> DF.PAC_NUM *****

```
1 *
2 *      DF      DF REQUEST  (GDPU -> ADPU)
3 *
4 *      All DF requests to the ARFs.
5 *
6 * 1) DF_FR1 - DF_FR5  FREQUENCY OF DF
7 *      Frequency consists of 10 BCD digits. The most
8 *      significant digit represents 1 GHZ.
9 *
10 * 2) DF_GAIN  RECEIVER GAIN SETTINGS
11 *      This field is ignored by ADPU software.
12 *
13 * 3) DF_BAND  BANDWIDTH VALUES
14 *      0 - 8 KHZ
15 *      1 - 15 KHZ
16 *      2 - 50 KHZ
17 *
18 * 4) DF_DET  DETECTOR VALUES
19 *      0 - FM
20 *      2 - AM
21 *      4 - CW
22 *      5 - LSB
23 *      6 - USB
24 *      7 - ISB
25 *
26 * 5) DF_TC  AGC TIME CONSTANT
27 *      This field is ignored by ADPU software
28 *
29 * 6) DF_DU  AGC DUMP CONSTANT
30 *      This field is ignored by ADPU software.
31 *
32 * 7) DF_SQ  SQUELCH
33 *      This field is ignored by ADPU software.
34 *
35 * 8) DF_CV  COARSE VERIFY FLAG
36 *      This flag is used only during calibration verify.
37 *      0 - fine DF
38 *      1 - coarse DF
39 *
40 * 9) DF_POL  ANTENNA POLARIZATION
41 *      This field is ignored by ADPU software.
42 *
43 * 10) DF_TYP  DF TYPE DESIGNATOR
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DF.PAC_NUM

```

44 *      1 - BITE
45 *      2 - MANUAL DF
46 *      3 - INTEROP
47 *      4 - AUTO DF (DS)
48 *      5 - GENERAL SEARCH (FINE)
49 *      6 - CALIBRATION
50 *      7 - AOI
51 *      8 - GEOSCREEN
52 *
53 * 11) DF_AUDIO      AUDIO CORRELATOR RECEIVER TUNING
54 *      0 - do not tune audio correlator receiver
55 *      1 - tune audio correlator receiver
56 *
57 * 12) DF_FINE      COARSE/FINE DF FLAG
58 *      0 - coarse DF
59 *      1 - fine DF
60 *
61 * 13) DF_RESP      RESPONSE FLAG
62 *      0 - no response
63 *      1 - send response
64 *
65 * 14) DF_CONT      CONTIGUOUS FLAG
66 *      0 - non contiguous DF
67 *      1 - contiguous segments---ie. do no re-tune receiver
68 *
69 * 15) DF_SEG      DF SEGMENT NUMBER
70 *      Valid segment numbers are from 1 thru 6.
71 *
72 * 16) DF_EN1,DF_EN2 ENABLE ARFX REPORTING
73 *      0 - Do not send additional DF data messages (UDR1,UDR2,UDR3)
74 *      1 - Do send additional DF data messages.
75 *
76 *      STRC
77 *      USES      1,DF_MID      MESSAGE ID ($12)
78 *      USES      1,DF_NIB      ARF ID. & BLOCK COUNT
79 *      BITS      4,DF_ARF      ARF ID.
80 *      BITS      4,DF_BLK      BLOCK COUNT
81 *      USES      2,DF_ACC      ACCOUNTABILITY
82 *      LABEL      DF_FRQ      FREQUENCY SEE NOTE 1
83 *      USES      1,DF_FR1      1ST 2 BCD DIGITS
84 *      USES      1,DF_FR2      2ND 2 BCD DIGITS
85 *      USES      1,DF_FR3      3RD 2 BCD DIGITS
86 *      USES      1,DF_FR4      4TH 2 BCD DIGITS
87 *      USES      1,DF_FR5      LAST 2 BCD DIGITS
88 *      USES      1,DF_GAIN      GAIN TYPE & MANUAL CNTRL
89 *      BITS      2,DF_TYPE      GAIN TYPE
90 *      BITS      6,DF_MAN      MANUAL GAIN SETTING

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DF.PAC_NUM

91	LABEL	DF_RCVR	RECEIVER R-924 CONTROL
92	USES	1,DF_RNG	BANDWIDTH AND DETECTORS
93	BITS	3,DF_BAND	BANDWIDTH CODE SEE NOTE 2
94	BITS	3,DF_DET	DETECTOR CODE SEE NOTE 3
95	BITS	1,DF_TC	AGC TIME CONSTANT
96	BITS	1,DF_DU	AGC DUMP CONSTANT
97	USES	1,DF_FLAGS	MISC FLAGS
98	BITS	1,DF_SQ	SQUELCH
99	BITS	1	
100	BITS	1,DF_EN1	ENABLE ARF1 REPORTING
101	BITS	1,DF_EN2	ENABLE ARF2 REPORTING
102	BITS	1,DF_CV	COARSE VERIFY FLAG
103	BITS	2,DF_POL	ANTENNA POLARIZATION
104	USES	1,DF_TYP	DF TYPE SEE NOTE 4
105	USES	1,DF_ASUB	DF SUB TYPE & AUDIO CORR.
106	BITS	1,DF_AUDIO	AUDIO CORR RCVR TUNE
107	BITS	1,DF_FINE	COARSE/FINE FLAG
108	BITS	1,DF_RESP	RESPOND FLAG
109	BITS	1,DF_CONT	CONTIGUOUS FLAG
110	BITS	4,DF_SEG	SEGMENT NUMBER
111	ENDST	DF	

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFF.PAC_NUM

13.2 DFF.PAC_NUM

***** Source Listing —> DFF.PAC_NUM *****

```

1 *
2 *           DFF      DF FREQUENCY FOR SIGNAL GENERATOR
3 *
4 * 1) FREQUENCY CONSISTS OF 10 BCD DIGITS. THE MOST
5 *   SIGNIFICANT DIGIT REPRESENTS 1 GHZ.
6 *
7 * 2) BANDWIDTH VALUES
8 *
9 *   0 - 8 KHZ
10 *   1 - 15 KHZ
11 *   2 - 50 KHZ
12 *
13 * 3) DETECTOR VALUES
14 *
15 *   0 - FM
16 *   2 - AM
17 *   4 - CW
18 *   5 - LSB
19 *   6 - USB
20 *   7 - ISB
21 *
22 *   DF TYPE VALUES
23 *
24 *   1 - BITE
25 *   2 - MANUAL DF
26 *   3 - INTEROP
27 *   4 - AUTO DF (DS)
28 *   5 - GENERAL SEARCH (FINE)
29 *   6 - CALIBRATION
30 *   7 - AOI
31 *   8 - GEOSCREEN
32 *
33 *   STRC
34 *   USES      1,DFF_MID      MESSAGE ID. ($13)
35 *   USES      1,DFF_TYP      DF TYPE SEE NOTE 4
36 *   USES      1,DFF_FR1      1ST 2 BCD DIGITS SEE NOTE 1
37 *   USES      1,DFF_FR2      2ND 2 BCD DIGITS
38 *   USES      1,DFF_FR3      3RD 2 BCD DIGITS
39 *   USES      1,DFF_FR4      4TH 2 BCD DIGITS
40 *   USES      1,DFF_FR5      LAST 2 BCD DIGITS
41 *   USES      1,DFF_RNG      BANDWIDTH & DETECTOR
42 *   BITS      3,DFF_BAND     BANDWIDTH CODE SEE NOTE 2
43 *   BITS      3,DFF_DET      DETECTOR CODE SEE NOTE 3

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFF.PAC_NUM

44

ENDST DFF

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFI.PAC_NUM

13.3 DFI.PAC_NUM

***** Source Listing —> DFI.PAC_NUM *****

```

1  *
2  *DFI - DF REQUEST FOR INTEROPERABILITY
3  *
4      STRC
5      USES      1,DFI_MID      MESSAGE ID ($21)
6      USES      1,DFI_NIB      PLATFORM ID & BLOCK COUNT
7          BITS  4,DFI_PLT      PLATFORM ID ($1)
8          BITS  4,DFI_BLK      BLOCK COUNT ($2)
9      USES      1,DFI_ACC      ACCOUNTABILITY
10     USES      1,DFI_SPR      UNUSED ($00)
11     USES      1,DFI_FR1      MSB OF FREQ (GHZ, 100MHZ)
12     USES      1,DFI_FR2
13     USES      1,DFI_FR3
14     USES      1,DFI_FR4
15     USES      1,DFI_FR5      LSB OF FREQ.
16     USES      1,DFI_MIS      PALLET SELECT & BW
17         BITS  4,DFI_PAL      PALLET SELECT (UNUSED 00)
18     ENDST      DFI

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFPR.PAC_NUM

13.4 DFPR.PAC_NUM

***** Source Listing --> DFPR.PAC_NUM *****

```

1  *
2  *          DFPR      DF RESPONSE  (GDPU -> PE)
3  *
4  *          MESSAGE TO TRANSFER DF RESPONSE INFORMATION
5  *          (TWO LOPS)
6  *
7  * 1) FREQUENCY CONSISTS OF 10 BCD DIGITS. THE MOST
8  *    SIGNIFICANT DIGIT REPRESENTS 1 GHZ.
9  *
10 * 2) DF ERRORS BIT MAP BIT ASSIGNMENTS
11 *
12 *    0 DF REJECTED (PREV. TOO LONG )
13 *    1 DF DID NOT COMPLETE IN SEGMENT
14 *    2 NO SIGNAL PRESENT
15 *    3 BASE LINE SIG PRES FAILED
16 *    4 BASE LINE SIG PRES ABORT
17 *    5 ROLL ANGLE EXCEEDS LIMIT
18 *    6 MULTIPLE MINIMA IN 1ST SEARCH
19 *    7 POOR FIT, BAD QUALITY
20 *    8 MISSING SEGMENT (FINE DF)
21 *    9 NO CAL TABLES
22 *
23 * 3) DF TYPE VALUES
24 *
25 *    1 - BITE
26 *    2 - MANUAL DF
27 *    3 - INTEROP
28 *    4 - AUTO DF (DS)
29 *    5 - GENERAL SEARCH (FINE)
30 *    6 - CALIBRATION
31 *    7 - AOI
32 *    8 - GEOSCREEN
33 *
34 *          STRC
35 *          USES      1,DFPR_MID      MESSAGE ID  ($14)
36 *          USES      1,DFPR_REJ      DF REJECTED FLAG
37 *          USES      2,DFPR_ACC      ACCOUNTABILITY
38 *          USES      1,DFPR_ACR      AUDIO CORRELATION FLAG (0 - FAIL)
39 *          USES      1,DFPR_CO      COUNTER FOR DS
40 *          USES      1,DFPR_POR      PORTION NUMBER  (1-14)
41 *          USES      1,DFPR_ENT      ENTRY NO. FOR DS (1-20)
42 *          USES      4,DFPR_SIG      SIGNAL TYPE BIT MAP
43 *          LABEL      DFPR_FRQ      DF FREQUENCY  SEE NOTE 1

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFPR.PAC_NUM

44	USES	1,DFPR_FR1	MSB OF FREQ. (BCD)
45	USES	1,DFPR_FR2	2ND BYTE OF FREQ.
46	USES	1,DFPR_FR3	3RD BYTE OF FREQ.
47	USES	1,DFPR_FR4	4TH BYTE OF FREQ.
48	USES	1,DFPR_FR5	LSB OF FREQ.
49	LABEL	DFPR_LP1	FIRST LOP
50	USES	1,DFPR_AR1	ARF ONE ID
51	USES	2,DFPR_ER1	ERROR CODE SEE NOTE 2
52	USES	1,DFPR_TY1	DF TYPE SEE NOTE 3
53	USES	1,DFPR_SS1	SIGNAL STRENGTH MEASUREMENT
54	USES	2	PAD FOR ALIGNMENT
55	USES	4,DFPR_LA1	LATITUDE (BAM)
56	USES	4,DFPR_LO1	LONGITUDE (BAM)
57	USES	2,DFPR_LB1	LINE OF BEARING (BAM)
58	USES	2,DFPR_QU1	QUALITY
59	USES	2,DFPR_HE1	HEADING (BAM)
60	USES	2,DFPR_RO1	AIRCRAFT ROLL (BAM)
61	USES	16,DFPR_BA1	5,6 OR 8 BASELINE PHASE ANLGES (BAM)
62	*		
63	USES	2	PAD FOR ALIGNMENT
64	LABEL	DFPR_LP2	SECOND LOP
65	USES	1,DFPR_AR2	ARF TWO ID
66	USES	1	UNUSED
67	USES	2,DFPR_ER2	ERROR CODE SEE NOTE 2
68	USES	1,DFPR_TY2	DF TYPE SEE NOTE 3
69	USES	1,DFPR_SS2	SIGNAL STRENGTH MEASUREMENT
70	USES	4,DFPR_LA2	LATITUDE (BAM)
71	USES	4,DFPR_LO2	LONGITUDE (BAM)
72	USES	2,DFPR_LB2	LINE OF BEARING (BAM)
73	USES	2,DFPR_QU2	QUALITY
74	USES	2,DFPR_HE2	HEADING (BAM)
75	USES	2,DFPR_RO2	AIRCRAFT ROLL (BAM)
76	USES	16,DFPR_BA2	5,6 OR 8 BASELINE PHASE ANGLES (BAM)
77	ENDST	DFPR	

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFR.PAC_NUM

13.5 DFR.PAC_NUM

***** Source Listing --> DFR.PAC_NUM *****

```

1 *
2 *      DFR      DF RESPONSE  (ADPU -> GDPU)
3 *
4 *      MESSAGE TO TRANSFER DF RESPONSE INFORMATION
5 *
6 * 1) DFR_ERR    DF ERRORS BIT MAP BIT ASSIGNMENTS
7 *      0 - DF REJECTED (PREV. TOO LONG )
8 *      1 - DF DID NOT COMPLETE IN SEGMENT
9 *      2 - NO SIGNAL PRESENT
10 *      3 - BASE LINE SIG PRES FAILED
11 *      4 - BASE LINE SIG PRES ABORT
12 *      5 - ROLL ANGLE EXCEEDS LIMIT
13 *      6 - MULTIPLE MINIMA IN 1ST SEARCH
14 *      7 - POOR FIT, BAD QUALITY
15 *      8 - MISSING SEGMENT (FINE DF)
16 *      9 - NO CAL TABLES
17 *
18 * 2) DFR_TYP    DF TYPE VALUES
19 *
20 *      1 - BITE
21 *      2 - MANUAL DF
22 *      3 - INTEROP
23 *      4 - AUTO DF (DS)
24 *      5 - GENERAL SEARCH (FINE)
25 *      6 - CALIBRATION
26 *      7 - AOI
27 *      8 - GEOSCREEN
28 *
29 * 3) DFR_SSM     SIGNAL STRENGTH MEASURE
30 *      This field contains the digitized AGC voltage which is a
31 *      crude measure of signal strength. Division of this field
32 *      by 1.5 should give an approximation of SNR in dB. This
33 *      field is only valid for calibration measurements.
34 *
35 * 4) DFR_LAT     LATITUDE
36 *      Latitude of ARF at time of measurement in BAM.
37 *
38 * 5) DFR_LON     LONGITUDE
39 *      Longitude of ARF at time of measurement in BAM.
40 *
41 * 6) DFR_LOB     LINE OF BEARING
42 *      Line of bearing to emitter with respect to ARF heading.
43 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFR.PAC_NUM

```

44 * 7) DFR_QUAL  QUALITY OF CAL TABLE FIT
45 *    Quality of the base line measurements cal table fit.  Valid
46 *    values are 0 thru 32 with 0 being a perfect fit to the cal
47 *    table.
48 *
49 * 8) DFR_HEA   HEADING
50 *    Heading of ARF at time of measurement
51 *
52 * 9) DFR_ROLL  ROLL ANGLE
53 *    Roll angle of ARF at time of measurement
54 *
55 * 10) DFR_BASE BASELINE PHASE MEASUREMENTS
56 *    Baseline phase measurements: 5 values for VHF LO, 8
57 *    values for VHF HI and 6 values for UHF.
58 *
59      STRC
60      USES      1,DFR_MID      MESSAGE ID  ($15)
61      USES      1,DFR_NIB      ARF ID & BLOCK COUNT
62      BITS      4,DFR_ARF      ARF ID
63      BITS      4,DFR_BLK      BLOCK COUNT
64      USES      2,DFR_ACC      ACCOUNTABILITY
65      USES      2,DFR_ERR      ERROR CODE SEE NOTE 1
66      USES      1,DFR_TYP      DF TYPE  SEE NOTE 2
67      USES      1,DFR_SSM      SIGNAL STRENGTH MEASURE
68      USES      4,DFR_LAT      LATITUDE      (BAM)
69      USES      4,DFR_LON      LONGITUDE      (BAM)
70      USES      2,DFR_LOB      LINE OF BEARING (BAM)
71      USES      2,DFR_QUAL     QUALITY
72      USES      2,DFR_HEA      HEADING        (BAM)
73      USES      2,DFR_ROLL     AIRCRAFT ROLL  (BAM)
74      USES      16,DFR_BASE    5,6 OR 8 BASELINE PHASE ANGLES (BAM)
75      ENDST      DFR

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFRJ.PAC_NUM

13.6 DFRJ.PAC_NUM

***** Source Listing --> DFRJ.PAC_NUM *****

```

1  *
2  *      DFRJ      DF REJECTED MESSAGE
3  *
4  * 1) DF TYPE VALUES
5  *
6  *      1 - BITE
7  *      2 - MANUAL DF
8  *      3 - INTEROP
9  *      4 - AUTO DF (DS)
10 *      5 - GENERAL SEARCH (FINE)
11 *      6 - CALIBRATION
12 *      7 - AOI
13 *      8 - GEOSCREEN
14 *
15      STRC
16      USES      1,DFRJ_MID      MESSAGE ID. ($16)
17      USES      1,DFRJ_MDR      MESS. ID. OF REQUEST
18      USES      2,DFRJ_ACC      ACCOUNTABILITY OF REQUEST
19      USES      1,DFRJ_TYP      DF TYPE SEE NOTE 1
20      USES      1,DFRJ_ASU      AUDIO CORR. & SUBTYPE
21      BITS      1,DFRJ_AUD      AUDIO CORRELATION
22      BITS      7,DFRJ_SUB      SUBTYPE
23      ENDST      DFRJ

```


Improved GUARDRAIL V MC68000 'DF' Files

14 FILES WITH EXTENSION '.R68'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA1.R68_NUM

14.1 DFA1.R68_NUM

***** Source Listing —> DFA1.R68_NUM *****

```

1 02002408DFA1****000008AADMINUC0000014E01BD
2 18003E08DFA4****0008AGSEND**0008DFA2****0008DSOB****0008GETPAK**00F2
3 18003E08PUTPAK**0008FMTFRQ**0008DFRPN**0008ARFID**0008?ACTIVE*FFF5
4 18002608??EXIT**0008??SCHED*0008??WRITW*007C
5 160018010000002808DFA1ST**003B
6 06002401000000040000000800000001C000000109D
7 22001C43000000070000000B0000000F5E
8 060010010000001400C015
9 06003C010000001A00000010001000080008000600022F0A3478000023EA005821
10 20001003000900000002D97
11 06003C010000003200000000245F3679000000007E057C017A0142844283207CB7
12 22000C43000000355A
13 2000104300000000003D50
14 06003C010000004A000000004E40247AFFAE4282343C0005102A000B0C390001D6
15 20001043000B00000004D35
16 0600400100000062000000006704343C00040500670A33FC00010000000060064279B1
17 20001C43000800000006500070000007796
18 060040010000007C00000000102A000D08000007670000A041EA000443FAFF7E4EB9F0
19 2000104300070000007F07
20 060040010000009600000000102A000A08C000010880000033C000000016122A000445
21 22000C43000000ABE4
22 20001043000600000099EE
23 06004001000000B0427900000018024100F0670833FC00010000001848E780004283D8
24 22001443000000B5000000C30F
25 06004001000000CA3679000000003E3C00062C7AFF2E428699CC9BCD08C3000008C3C2
26 200010430003000000CFBB
27 06004001000000E4000F207C000000004E404CDF000108C0000033C000000016323C31
28 22000C43000000FB94
29 20001043000C000000EB96
30 06004001000000FE011D51C9FFFE42833679000000003E3C00062C7AFEF2428699CCCF
31 2000104300030000010B7E
32 06004001000001189BCD08C3000008C3000F207C000000004E403679000000007E023A
33 20001C43000C0000012700020000012F1B
34 06003801000001327C007A01283AFEC84283207C000000004E404284207C1E
35 20001043000B0000014140
36 0600180100000148000000004E400A
37 20001043000A0000014B37
38 04000E00010C000000ED

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.R68_NUM

14.2 DFA2.R68_NUM

***** Source Listing —> DFA2.R68_NUM *****

```

1 02002408DFA2****000008ADFUC***00000726011B
2 18003E08DFDIB***0008GETBITS*0008DETBAND*0008GETLPAK*0008PUTPAK**0008
3 18003E08CLRACCS*0008INITDF**0008STARTDF*0008WAITDF**0008FMTFRQ**00E2
4 18003E08SIOLOAD*0008SIOSEND*0008RFPSETUP0008ACUSETUP0008NQUEUE**0044
5 18003E08DQUEUF**0008ADJFDFC*0008DMPFIFO*0008ACCBUF**0008ACCUM***004F
6 18003E08SIGPRES*0008DETSIG**0008READAGC*0008ABTON***0008ABTOFF**0022
7 18003E08SEQTAB**0008SEQTAB2*0008TUNSEQ*0008TUNEACU*0008TUNEFDFC00A2
8 18003E08DFA5****0008DFA6****0008DFA2DAT10008DFA2OUT10008DFA2MSK100C3
9 18003E08DFA2DAT20008DFA2OUT20008DFA2MSK20008NAVDATA*0008CALIB***004D
10 18002608?ACTIVE*FF08??EXIT**0008??SCHED*00BA
11 160040010000000408RESET***000000001808RDATA***000000001C08TDATA***0026
12 160040010000000508RESOVF**000000000A08ADFBT***000000000208NCOMPL**0017
13 1600400100000003208BNDWDTH*000000002008SENDID**000000001208SKFRST**0090
14 160040010000000608WAITING*000000001608RMSG****00000000008NPEND***0014
15 1600400100000002608RFPDPOS000000001408DFBOX***000000002808RFPPTR**0076
16 160040010000000C08SIGNAL**00000000D808INTCPU**000000002408BAND****0043
17 16002C01000000DA08DFA2E***000000001608RID*****0045
18 0600100100000000E0001DA
19 0600100100000002C00536A
20 0600100100000003800C0F1
21 0600280100000003E00000010001000080008000600025B
22 060034010000000D40000000000002F0A34780000226A0058245F267906
23 200010030028000000DFC6
24 060038010000000E80000000028790000000036390000000087533893267985
25 200028430020000000EB0021000000F10022000000F73F
26 06003C01000000FE000000002879000000003639000000008753389333FC0001DA
27 2000284300230000010100240000010700250000010DF1
28 06003C0100000116000000D852B9000000D44279000000084279000000004279B6
29 22001C43000001190000011F000001251F
30 2000104300270000012B3A
31 060040010000012E0000000A42790000000C42B90000002E4279000000121029000C7E
32 2200244300000131000001370000013D000001438B
33 06003801000001480C000001664C1029000D080000066642323C03004EB945
34 060040010000015E000000000C41002661245F90000000033FCFFFF0000000A6000BE
35 22000C430000017519
36 20001C4300010000016100170000016D99
37 0600400100000178011A0C410003661245F90000000033FCFFFF0000000A6000010285
38 22000C430000018D01
39 20001043001800000185EF
40 060040010000019233FC70010000000A41E900044EB90000000033C00000002434006C
41 2200144300000199000001A943
42 200010430002000001A3E7
43 06003801000001AC1029000A323C02054EB90000000033C1000000324EB928

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.R68_NUM

```

44 22000C43000001BFCF
45 200010430001000001B9D2
46 06004001000001C2000000001D42000D1D69000C00013D69000200021D69000D0007B3
47 200010430003000001C5C4
48 06004001000001DC1D69000400081D69000500091D690006000A1D690007000B1D6902
49 06004001000001F60008000C426E000423CE0000002E41EE000E23C8000000504EB95C
50 22001443000002030000020D73
51 0600400100000210000000001229000D0801000666144A790000000A6706343C000824
52 22000C43000002236A
53 2000104300050000021373
54 060040010000022A6034343C0007602E1029000D323C03004EB900000000340133FCD2
55 2000104300010000023F4B
56 06003C010000024400010000000C0C290006000C660E0C410006660833FC0001BE
57 22000C430000024944
58 060040010000025C00000000303AFDC25340C1FC007045F9000000000C290006000CED
59 20001C4300270000025F00190000026F6F
60 060040010000027666101229000B08010003670645F900000000D4C05342C5FC000ED6
61 20001043001A00000287EA
62 0600400100000290D4C24EB9000000003C2A0000670004122C6A00022A6A0006286AE3
63 20001043000600000297EE
64 06004001000002AA000A1029000D0800000466000118264943FAFD7841EB00044EB9DA
65 06003C01000002C40000000042790000003C102B0004024000F0670833FC0001F0
66 22000C43000002CDC0
67 200010430009000002C7BB
68 06003C01000002DC0000003C102B000A08C00001088000000280FFFFFFE333C0B8
69 22000C43000002DFAE
70 06004001000002F40000003A41FAFD46303C0006484042414EB900000000303C00001B
71 22000C43000002F796
72 20001043000A0000030977
73 060040010000030E4EB90000000041F900000000323AFD0853413401C3FC000CD0C1D1
74 20001C43000B00000313001B0000031929
75 060040010000032823C80000002824484279000000264EB900000000303C08004EB9AC
76 220014430000032D000003351F
77 20001043000C0000033B43
78 060040010000034200000003202C3FC000441F900000000D1C14EB900000000E54283
79 20002843000B00000345001C00000351000D0000035949
80 060040010000035C41F900000000203020004EB9000000004EB90000000048E780C033
81 20002843001D0000036100100000036B000700000371FB
82 060040010000037620790000000022790000000303900000000B15032904CDF0301B1
83 2000284300230000037B0024000003810025000003877D
84 0600400100000390204A4840323C01044EB900000000303C00014EB9000000003E0008
85 20001C4300010000039D000E000003A728
86 06004001000003AA33FC000100000008102B000D0800000667000084301EE340DCC086
87 22000C43000003B1DB
88 06004001000003C4301DE340DAC0588C6072427900000026264923CE000000284EB9C2
89 22001443000003D3000003DBD3
90 06004001000003DE00000000303C08004EB900000000204D4EB9000000002A48201C3B

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA2.R68_NUM

91 20002843000C000003E1000B000003EB000D000003F389
92 06003C01000003F84EB9000000004EB90000000048E780C0207900000000227911
93 200028430010000003FD00070000040300230000040D23
94 060040010000041000000000303900000000B15032904CDF03014840323C01044EB948
95 20001C4300240000041300250000041904
96 060040010000042A00000000303C0001204E4EB9000000002C7AFBEC3E00204E23CE7F
97 20001C4300010000042D000E0000043904
98 0600400100000444000000282C4B122B000D4EB9000000002C48BE46670001FC30066F
99 22000C430000044744
100 2000104300040000045332
101 06003C010000045E53405340204D4EB9000000004EB90000000051C8FFF223CCC1
102 20001C43000C00000469000D0000046F88
103 06004001000004760000004C4279000000104A790000000E674C4E715247BE466E00DA
104 22001C43000004790000047F00000485F6
105 060038010000049001C6287AFBB8201C23CC0000004C4EB9000000004EB98C
106 22000C430000049DEE
107 200010430010000004A3D6
108 06003801000004A6000000004840323C01044EB9000000003007204E4EB969
109 20001C430007000004A90001000004B513
110 06004001000004BC00000000301EE340DCC04A790000000E6606BE466C0001844EB9B3
111 22000C43000004CBC0
112 20001043000E000004BFBC
113 06003801000004D6000000004EB9000000004A79000000086700009C427957
114 22000C43000004E5A6
115 20001C430008000004D9000F000004DFAA
116 06004001000004EC000000084A790000000C660000E042423010360045FAFB524EB91F
117 22001443000004EF000004F59B
118 0600400100000506000000004BEA00104EB9000000004A406704343C0001508D4EB918
119 20001C4300110000050900150000051335
120 06003C0100000520000000004A4066044242600852420C4200026C24508D4EB960
121 2000104300150000052350
122 06003C010000053800000000D4400C42000267340C420001660000FC33FC0001A0
123 2000104300150000053B38
124 060040010000055000000012600000F033FC00010000000C2848264A207AFAEA5343D2
125 22001443000005530000055FCB
126 060040010000056A4EB90000000051CBFFF8600000D033FC00010000000C600000C4A0
127 22000C430000057F0B
128 2000104300120000056F07
129 06004001000005844A790000000066624A790000000C66444A79000000106600009E55
130 22001443000005910000059953
131 20001043002700000589D8
132 060038010000059E45FAFAB430104EB90000000022484EB9000000004A79B6
133 20001C430011000005A90014000005B1F8
134 06004001000005B40000000C660C33FC000100000010600000825343207AFA864EB9A9
135 22001443000005B7000005C105
136 06003801000005CE0000000051CBFFF8606E2248207AFA74361953434EB9AF
137 200010430012000005D1A5

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA2.R68_NUM

138 06004001000005E40000000051CBFFF860580C40000267080C4000046720604A4EB9C0
139 200010430013000005E78E
140 06004001000005FE00000000247AFA2A024000FFB07AFA206E04303AFA1A1540000624
141 2000104300160000060170
142 0600400100000618602C4EB900000000247AFA0C122A0006024000FF024100FF9041CE
143 2000104300160000061F52
144 060038010000063215400006600E4E71301845FAFA164EB9000000004A79A0
145 2000104300110000064531
146 060038010000064800000000E6600FE3C52476000FE3E4EB9000000004EB982
147 22000C430000064B3E
148 2000104300080000065B24
149 060040010000065E0000000004A790000000A6B0000964A790000000C67162248361982
150 220014430000066700000671A3
151 20001043000F0000066117
152 0600400100000678207AF9D653434EB90000000051CBFFF8602E301845FAF9C64EB947
153 20001043001300000683F1
154 0600400100000692000000004AB900000002E67622C7AF98E302E000408C000023D4051
155 22000C430000069BEE
156 20001043001100000695E1
157 06004001000006AC000460082C494EB9000000004AB90000002E67402C7AF96C102EFE
158 22000C43000006BDCC
159 200010430004000006B7CC
160 06004001000006C6000708000006671A3679000000007E0242864285280E4283207C02
161 20001043001E000006D396
162 06004001000006E0000000004E4060183679000000007E0242864285280E4283207C78
163 20001C43002A000006E3001F000006ED5C
164 06004001000006FA000000004E40207900000000227900000000303900000000B1508D
165 20003443002A000006FD00200000070500210000070B002200000711A3
166 060030010000071432904279000000D84284207C000000004E4069
167 22000C430000071B6D
168 200010430029000007233A
169 04000E000100000000ED

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA4.R68_NUM

14.3 DFA4.R68_NUM

***** Source Listing —> DFA4.R68_NUM *****

```
1 02002408DFA4****000008AADMINUC0000006601A3
2 18003208NAVDATA*0008DNAV****0008??EXIT**0008??READW*0061
3 160018010000001808DFA4ST**0048
4 0600400100000014000000000428336790000000003E3C00052C7AFFEE7C0099CC9BCDD6
5 22000C430000001778
6 2000104300010000001F6D
7 060040010000002E08C3000708C3000F207C0000000004E4041F900000000217AFFBA27
8 20001C43000300000003B000000000004300
9 06003C01000000480000217AFFB80004317AFFB60008317AFFB8000A4284207CE9
10 06001801000000600000000004E40F3
11 20001043000200000006328
12 04000E0001000000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA5.R68_NUM

14.4 DFA5.R68_NUM

***** Source Listing —> DFA5.R68_NUM *****

```

1 02002408DFA5****000008ADFUC***0000033C0106
2 18003E08DFA6****0008CPUSEND*0008NAVDATA*0008GETLPAK*0008PUTLPAK*0000
3 18003E08GETPAK**0008GETBITS*0008?ACTIVE*FF08??EXIT**0008??SCHED*0082
4 16001801000000A808DFA5E***00E3
5 06001601000000080000000000DB
6 06004001000000A20005000508062F0A34780000206A0058245F10280007323C030005
7 200010030007000000AD19
8 06004001000000BC4EB9000000000C280006000166684E710C010006663243FAFF2C1B
9 200010430006000000C1C6
10 06004001000000D61368000600061368000800C81368000900091368000A000A13683A
11 06004001000000F0000B000B1368000C000C336800040004600002280C010005662853
12 060040010000010A4E71303C000443FAFEE45E9000847E80008B70A660651C8FFFAAA
13 06003C0100000124600C3029000408C0000833400004B27AFF6E66000176537946
14 060040010000013C000000A20C4100056600008243FAFEB613680001000113680007B0
15 22000C430000013F4F
16 060040010000015600071368000800081368000900091368000A000A1368000B000B23
17 06004001000001701368000C000C1368000D000D1028000D488045FAFF20143200006F
18 060040010000018A488245E8000EE542534247E9000E26DA51CAFFFC2368008E008E72
19 06004001000001A42368009200920C2800060001660C30280004816900046000016E9F
20 06004001000001BE3368000400046000016443FAFE3610280001B0290001660000D8D0
21 06004001000001D845E9000847E80008303C0004B70A660000C651C8FFF81028000DC1
22 06004001000001F2488045FAFEAE103200004880E540534045E8000E47E9000E241A9A
23 060040010000020CD59B51C8FFFA2428008ED5A9008E24280092D5A9009230280004F9
24 0600400100000226816900041368000700070C41000366244E7145F900000000236AB6
25 2000104300020000023D4C
26 060040010000024000000096236A0004009A336A0008009E336A000A00A0600000D2FA
27 060040010000025A0C410001660000CA4EB900000000303C00A25340244E14D951C8BF
28 2000104300030000026721
29 0600400100000274FFFC3D68000200021D7C004B000048E700803679000000007E02DD
30 2000104300000000028B00
31 06003C010000028E42864285280E4283207C000000004E404CDF010033FC000519
32 2000104300090000029BE7
33 06003C01000002A6000000A2607E4E7133FC0005000000A20C410005660A537972
34 22001443000002A9000002B525
35 06004001000002BE000000A26000FE84303C000443FAFD3443E90008421951C8FFFCF4
36 22000C43000002C1CC
37 06004001000002D80C410001664C4E714EB9000000001D7C001500001D7C00140001BD
38 200010430005000002E5A1
39 06004001000002F23D68000200023028000408C000083D4000041D680001000648E7B4
40 060040010000030C00803679000000007E0342864285280E4283207C000000004E4046
41 20001C430001000003130009000003233B
42 06003801000003264CDF01002C484EB9000000004284207C000000004E4001
43 20001C4300040000033100080000033905

```


Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA5.R68_NUM

44 04000E000100000000ED

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA6.R68_NUM

14.5 DFA6.R68_NUM

***** Source Listing --> DFA6.R68_NUM *****

```

1 02002408DFA6****000008ADFUC***000004C6017A
2 18003E08NAVDATA*0008MAXROLL*0008ERROR2**0008SCMIN***0008SCVAL***0002
3 18003E08DFA6DAT10008DFA6OUT10008DFA6MSK10008DFA6DAT20008DFA6OUT20023
4 18003E08DFA6MSK20008DFA6DAT30008DFA6OUT30008DFA6MSK30008DFA6DAT40028
5 18003E08DFA6OUT40008DFA6MSK40008DFRPEN**0008GETPAK**0008BLSGPRS*00E3
6 18003E08INPQUAD*0008DIRREV**0008BCDBIN**0008COARSE**0008P2SRCH**0016
7 18003E08SQTHRESH0008CPUSEND*0008PUTLPAK*0008BITETBL*0008?ACTIVE*FF58
8 18001A08??EXIT**0008??SCHED*006B
9 160040010000005008QTHRESH*0000000004C08QUAL****0000000006008DFA6E***0056
10 1600400100000002408BADBL***0000000003408NORES***0000000003E08IFRQMHZ*0094
11 1600400100000003A08FRHPTR**0000000005408NORES2**0000000003208MINX1***005D
12 1600400100000004808MINX2***0000000004208ROLL****0000000002C08THETA***00AB
13 1600400100000003608FRLPTR**0000000004A08LOB*****0000000004008BANDLB**00A1
14 1600400100000003008MINVAL1*0000000002A08NBL*****0000000002608INTFRQ**00C2
15 1600180100000004408MINVAL2*00FF
16 060014010000000500000000194
17 06001E01000000056071C0102060005080646
18 060028010000000602F0A3478000002A6A0058245F26797E
19 20001003001D0000000654B
20 0600400100000006E0000000002879000000003C390000000008D533893102D000708003E
21 20002843000500000007100060000007700070000007DFE
22 060040010000000880005670004124EB9000000001D7C001500001D7C000400013D6DB2
23 200010430012000000093E8
24 06004001000000A2000200023D6D000400041D6D000100061D6D00060007302D0004D8
25 06004001000000BC0240000466000266102D00014239000000000C0000016616102DCA
26 22000C430000000CDC2
27 06004001000000D6000708000006660C13FC000100000000600000C641ED000E102DAD
28 22000C430000000E5AA
29 06003801000000F0000D48804241142D0007080200066704323C00012679A8
30 06003801000001060000000002879000000003C390000000008D5338934EB9F2
31 2000284300080000010900090000010F000A000001152A
32 060040010000011C00000000BD5338930C4100026610302D000408C000043D4000044E
33 2000104300130000011F5A
34 0600380100000136600001F04A41670C302D000408C000033D40000433C299
35 060040010000014C000000024102D000708000006662043F9000000002D690000000896
36 22000C430000014F3F
37 2000104300000000015F2D
38 06004001000001662D690004000C3D69000800143D69000A001660182D6D0096000874
39 06004001000001802D6D009A000C3D6D009E00143D6D00A00016302E00166A024440D8
40 060040010000019AB079000000006F0C302E000408C000053D400004102D000D4880B8
41 2000104300010000019FEC
42 06004001000001B433C000000004043FAFE9F1A310000488533C50000002A43FAFE364C
43 22001443000001B9000001C903

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.R68_NUM

```

44 06004001000001CE204D4EB90000000045EE001823CA0000002C4EB900000000302EAD
45 22000C43000001DFAF
46 20001C430014000001D50015000001E59C
47 06004001000001E80004024000246600013841ED0008303C000A4EB90000000023C130
48 200010430016000001FF77
49 0600380100000202000000264A3900000000660000C22679000000002879AC
50 22001443000002050000020B73
51 20001043000B000002156B
52 0600400100000218000000003C39000000008D5338934EB900000000BD5338934A391A
53 20002843000C0000021B000D0000022100170000022BD8
54 0600400100000232000000346714123AFDFA4881302E000403C03D400004600000DEE6
55 22000C430000023558
56 060040010000024C41ED0008303C00064EB90000000006810000003283FC006433C12C
57 200010430016000002591C
58 06004001000002660000003E4EB90000000033EE001600000042267900000000287953
59 220014430000026900000277A3
60 20001C4300190000026F000E0000027D6A
61 0600400100000280000000003C39000000008D5338934EB900000000BD5338934A39B2
62 20002843000F0000028300100000028900180000029399
63 060040010000029A00000054670E4E71302E000408C000073D4000043D7AFD9A001085
64 22000C430000029DF0
65 06004001000002B4203AFD966A02448042414A806706E280524160F63D410012605A97
66 06004001000002CE4E71424441F900000000424C323C0005243AFD46B4986708524087
67 20001043001C000002D798
68 06004001000002E851C9FFF86032EB4045F900000000D5FC0C000018D4C0303AFD2AB5
69 20001043001C000002F57A
70 06004001000003025340320043EE00183419945A6A024442B45A6E063601964007C41F
71 060040010000031C51C8FFEC3D4400044246604442464A7900000000673A4E71102EFC
72 2000104300110000032F4A
73 06004001000003360006323C000241FAFD1AB018670A4E7151C9FFF860204E713C3CF8
74 06004001000003500001284E4EB900000000323C00285341204E224C10D951C9FFFCCE4
75 200010430012000003591F
76 06003C010000036AC94E48E7020C3679000000007E0442864285280E4283207CA5
77 20001043001A00000375FB
78 0600400100000382000000004E404CDF30404A466700010E197C004C0000322C0002C4
79 20001043001F00000385E6
80 060040010000039C0041C0003941000248E700043679000000007E0442864285280C76
81 20001043001A000003ADC3
82 06003C01000003B64283207C000000004E404CDF20004EB900000000284E4EB946
83 20001C43001F000003BD0012000003C9C4
84 06004001000003CE00000000197C004D0000197C00040001322D00020041C000394190
85 200010430012000003D1A7
86 06004001000003E8000241F90000000043EC0004323C000822D851C9FFFC48E7008625
87 200010430002000003EF99
88 06004001000004023679000000007E0442864285280C4283207C000000004E404CDFA5
89 20001C43001A00000407001F0000041722
90 060040010000041C61001D7C004E00001D7C00040001322D00020041C0003D410002D1

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA6.R68_NUM

91 060040010000043643EE0004323C000322D851C9FFFC43EE001441ED0008323C0004DD
92 060040010000045012D851C9FFFC303AFBD2907AFBC83D40001A3D7AFBCE001C3D7A78
93 060040010000046AFBC6001E3D790000000000203D7900000000002248E700043679DC
94 20001C4300030000047300040000047B84
95 0600400100000484000000007E0442864285280E4283207C000000004E404CDF2000B0
96 20001C43001A00000487001F0000049722
97 060038010000049E2C4D4EB9000000002679000000002879000000003C39EA
98 20002843001B000004A50005000004AB0006000004B142
99 06003001000004B400000000BD5338934284207C000000004E4046
100 20001C430007000004B7001E000004C3DA
101 04000E000100000000ED

Improved GUARDRAIL V MC68000 'DF' Files
DRG1:[ALGO.IGR.MC68000_DF]DFBANDS.R68_NUM

14.6 DFBANDS.R68_NUM

***** Source Listing —> DFBANDS.R68_NUM *****

```
1 02002408DFBANDS*000008ACCDs***0000000A01F7
2 1600400100000000608UHF*****000000000208VHFHI***000000000008MAXROLL*003C
3 060020010000000000C00000000000000000D9
4 04000E000100000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOL.R68_NUM

14.7 DFCOL.R68_NUM

***** Source Listing —> DFCOL.R68_NUM *****

```
1 02002408DFCOL***000008ADFUC***000001C40155
2 160040010000015C08VHICOAF*00000000C808VHICOAR*00000000C2408TUNEACU*0095
3 160040010000008208VLOCOAA*00000001B208UHFCOAA*000000011208VHICOAA*00C0
4 160040010000000BC08VLOCOAF*000000003A08VLOCOAR*000000002E08TUNEFDFC00A2
5 160040010000000008TUNESQ*00000001B808UHFCOAF*000000016808UHFCOAR*00B4
6 060040010000000000051C001C0010009000250900051C001C0010009000250900059E
7 0600400100000001A1C001C000F008F00250900010000000100000000157D005A157D1B
8 06004001000000034005A057D005A000F138113010182010288830803978417048B85B6
9 0600400100000004E0B0525091301138101020182000808038883170497840B058B858B
10 0600400100000006825091381000A13010182010288830803978417048B850B05250951
11 06004001000000082000F861082008610820082008200820082008200820082008200E8
12 0600400100000009C861082008610000882008200820082008200820082008610000340
13 060040010000000B6820086108200FF310056882100563A210056000F13011381130267
14 060040010000000D0138201030183010401841305138508068886170797870B0800081F
15 060040010000000EA8B882509138113011382130201830103000B01840104138513056F
16 060040010000010488860806978717078B880B082509000F514055502CB028A051408E
17 060040010000011E55502CB028A04D304920410041004100410041000008410082005B
18 06004001000001385550514028A02CB055505140000B28A02CB049204D304100410059
19 060040010000015241004100410041008200FF31005488210054BB210052000F848519
20 060040010000016C0405978617068F810F01888208028983090393841304848504057D
21 060040010000018697860008170625090F018F810802888209038983000B130493843D
22 06004001000001A00405848517069786040584851706978625090000000000000F3112
23 06002001000001BA0054082100540B210052CF
24 04000E000100000000ED
```

AD-A184 057

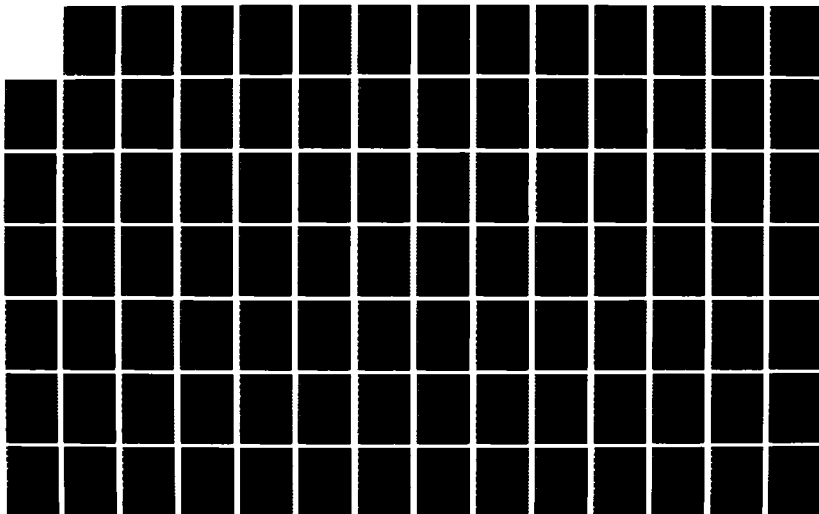
US ARMY INTELLIGENCE CENTER AND SCHOOL (USAICS)
SOFTWARE ANALYSIS AND MAN (U) JET PROPULSION LAB
PASADENA CA B PARDO 05 MAR 87 JPL-D-4216 NAS7-918

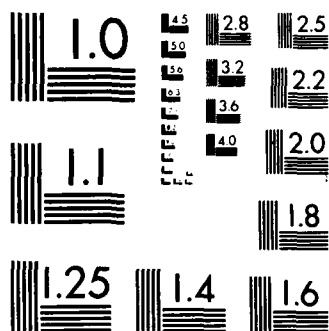
2/5

UNCLASSIFIED

F/G 12/5

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOL2.R68_NUM

14.8 DFCOL2.R68_NUM

***** Source Listing —> DFCOL2.R68_NUM *****

```
1 02002408DFCOL2**000008ADFUC***000001920184
2 1600400100000018208UHFCOAF20000000012E08UHFCOAR20000000011E08VHICOAF20021
3 1600400100000008608VHICOAR20000000004A08VLOCOAA20000000017A08UHFCOAA200A3
4 16004001000000D208VHICOAA20000000007608VLOCOAF2000000000008VLOCOAR20093
5 060040010000000000000000C138113010182010288830803978417048B850B052509E5
6 060040010000001A1301000B13810102018208038883170497840B058B852509138138
7 0600400100000034000A13010182010288830803978417048B850B0525090000000443
8 060040010000004E8610820086108200000B86108200861082008200820082008200F8
9 060040010000006882008200861000038200861082000000000004C310056BB21005615
10 06004001000000823A21005600000000C130113811302138201030183010401841305FE
11 060040010000009C138508068886000B170797870B088B88250913811301138213027C
12 06004001000000B601830103000B018401041385130588860806978717078B880B08B8
13 06004001000000D0250900000000C514055502CB028A0514055502CB028A04D30492015
14 06004001000000EA41004100000B410041004100410082005550514028A02CB055503D
15 06004001000001045140000B28A02CB049204D30410041004100410041004100820086
16 060040010000011E00000000CC310054BB210054BB21005200000000C848504059786B0
17 060040010000013817068F810F01888208028983090393841304000B848504059786AF
18 0600400100000152170625090F018F810802888209038983000B130493840405848584
19 060040010000016C1706978604058485170697862509000000000000000000000098
20 06002401000001860C3100540B2100540B210052BF
21 04000E000100000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFCOM.R68_NUM

14.9 DFCOM.R68_NUM

```
***** Source Listing --> DFCOM.R68_NUM *****
*****
1  02002408DFCOM***000008ACCDs***00000010010E
2  1600400100000000C08CALIB***000000000E08DFRPEN**000000000008NAVDATA*002A
3  04000E000100000010DD
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFFEED.R68_NUM

14.10 DFFEED.R68_NUM

```
***** Source Listing --> DFFEED.R68_NUM *****
*****
1  02002408DFFEED**000008GADMINUC0000003E0199
2  18003E08REPIP***0008RELPAK**0008DFRPT***0008?ACTIVE*FF08??EXIT**00D8
3  18000E08??SCHED*002D
4  1600180100000000008DFFEEDST0034
5  060028010000000002FOA347800002C6A0058245F4A79B8
6  200010030003000000005C5
7  06003C0100000000E00000000661A3679000000007E0742864285280E4283207CD5
8  20001C430000000000011000200000001955
9  06003C01000000026000000004E4060064EB9000000004284207C000000004E40AC
10 200028430005000000029000100000033000400000003BD4
11 04000E000100000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG1.R68_NUM

14.11 DFG1.R68_NUM

***** Source Listing —> DFG1.R68_NUM *****

```

1 02002408DFG1****000008GADMINUC0000014A01B5
2 18003E08GETNUM**0008GETPTR**0008REMBOT**0008PUSH****0008ACCOUNT*00E4
3 18003E08PUTPAK**0008HBL0K***0008DFG2*****0008HBDDB*****0008GETBITS*008D
4 18003E08PUTBITS*0008?ACTIVE*FF08??EXIT**0008??SCHED*0008??LOCKW*0023
5 18000E08??UNLOC*0013
6 16002C010000002408DFG1E***000000000408DFTYPTAB0097
7 060034010000000400000000000000120000002400000036000000480D
8 2000404300080000000700080000000B00080000000F000800000013000800000017EA
9 06002C01000000180000005A00000006C0000006C2F0A34789E
10 2000284300080000001B00080000001F00080000002300
11 0600280100000028000023EA005800000000245F367912
12 22000C43000000315E
13 20001003000B0000002999
14 0600400100000036000000007601207C000000004E40267AFFBA102B000A323C0205CF
15 20001C43000600000039000E00000041F3
16 06004001000000504EB9000000000C0100026F16143C0002323C0205102B000A4EB9BB
17 200010430009000000552F
18 060040010000006A000000001740000A4EB900000000102B000C48800C4000016D001E
19 20001C43000A0000006D00000000007793
20 060040010000008400820C4000086E7A4E714EB900000000B1FC00000000661A340050
21 20001043000100000093F9
22 060040010000009E5342E54245FAFF60207220004EB9000000004EB900000000142BC2
23 20001C430002000000AF0001000000B51A
24 06003C01000000B800000C020017660A4EB900000000374600022248204B4EB90E
25 200010430004000000C5C4
26 06004001000000D0000000000C400001662C142B000D0802000666220C2B00170000D8
27 200010430003000000D3B7
28 06004001000000EA661A286B0018B9FC00000000670E2C4B4EB900000000264C60002A
29 200010430005000000FF89
30 0600380100000104FF442C4B4EB90000000048E74000367900000000207C41
31 20001C4300050000010D00060000011750
32 060040010000011A000000004E404CDF00024A4166183679000000007E047C007A01B2
33 20001C43000F0000011D00070000012D1F
34 060038010000013442844283207C000000004E404284207C000000004E40E7
35 20001C43000D0000013D000C00000147E2
36 04000E000100000000ED

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG2.R68_NUM

14.12 DFG2.R68_NUM

***** Source Listing --> DFG2.R68_NUM *****

```
1 02002408DFG2****000008GADMINUC0000006A0195
2 18003E08DFG2****0008DFTYPTAB0008HBLOK***0008FINDOLD*0008??EXIT**000C
3 18002608??LOCKW*0008??SCHED*0008??UNLOC*0088
4 16002C010000000008SUM*****000000000208DFG2E***0017
5 06004001000000024279000000003679000000007601207C000000004E40303C000838
6 22000C430000000788
7 20001C4300020000000D00050000001558
8 06003C010000001C5340534043F90000000020594EB90000000051C8FFF6367902
9 20001C4300010000002500030000002D2B
10 060038010000003400000000207C000000004E404A79000000006718367972
11 22000C43000000454A
12 20001C4300020000003700070000003D04
13 060040010000004A000000007E047C147AFF42844283207C000000004E404284207CCD
14 20001C4300000000004D00060000005DD1
15 0600180100000064000000004E40EF
16 2000104300040000006722
17 04000E000100000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG3.R68_NUM

14.13 DFG3.R68_NUM

***** Source Listing —> DFG3.R68_NUM *****

```

1 02002408DFG3****000008GADMINUC000009C40131
2 18003E08GETBITS*0008PUTBITS*0008GETFINE*0008GETCRS**0008GETPAK**00B5
3 18003E08GETDF***0008DFPE2***0008ARDFBITR0008ADFCOL**0008GSDF****008D
4 18003E08A0IDF***0008GEOCOL**0008GASEND**0008HBL0K***0008DFTYPTAB003A
5 18003E08DFG3DAT10008DFG3OUT10008DFG3MSK10008DFG3DAT20008DFG3OUT20014
6 18003E08DFG3MSK20008TFLAG1**0008TFLAG2**0008DPUMODE*0008??EXIT**00FC
7 18001A08??LOCKW*0008??UNLOC*0041
8 160040010000003808ACORREQ*000000003908ACORAV**000000000008CURDF***00AF
9 160040010000006C08DFG3****000000003A08SIGREQ**000000003B08INTEROP*004F
10 0600320100000034000000000001000000000405040404040404046D
11 0600340100000048000000000000000000000000000000000000000000007D
12 2000404300060000004B00070000004F00060000005300060000005700080000005B9D
13 060038010000005C0000000000000000000000000000000000000048E7FFFE2079A0
14 2000344300090000005F000600000063000A00000067000B0000006BB1
15 060038010000007200000000227900000000303900000000815032904A3935
16 20002843000F0000007500100000007B001100000081D4
17 06004001000000880000003B660008F44A3900000034670003FC4A3900000035670058
18 22001C430000008B000000950000009FC0
19 06004001000000A200D43679000000007601207C000000004E404EB90000000041FAB1
20 20002843000D000000A90019000000B10002000000B93A
21 06004001000000BCFF443169000000001169000200023169000400041169000600067A
22 06004001000000D61169000700071169000800081169000900091169000A000A11693D
23 06004001000000F0000B000B1169000C000C1169000D000D1169000E000E1169000F6E
24 060040010000010A000F11690010001011690011001121690012001211690016001615
25 06003C010000012411690017001711690018001811690019001948E7008036793C
26 060040010000013C00000000207C000000004E404CDF010010280007323C03004EB96F
27 20001C43000D0000013F001A00000145D4
28 06003C0100000156000000000C4100036600062C13FC00000000003413FC00002C
29 22000C430000016925
30 2000104300000000015933
31 060040010000016E00000035600006184A3900000036670000D041FAFE981028000797
32 22001443000001710000017B99
33 0600400100000188323C03004EB90000000053413401323C03004EB9000000000C4229
34 20001C4300000000019100010000019F4E
35 06004001000001A20005660608800004600408C000040C420001661408C0000513FC44
36 06004001000001BC00000000003613FC0000000000341140000743FAFE303368000025
37 22001443000001C1000001C9FB
38 06004001000001D60000136800020002336800040004136800060006136800070007B0
39 06004001000001F01368000800081368000900091368000A000A1368000B000B136815
40 060040010000020A000C000C1368000D000D1368000E000E1368000F000F1368001045
41 06004001000002240010136800110011236800120012136800160016136800170017E7
42 060040010000023E136800180018136800190019600005404A390000003C6700017CD9
43 22000C43000002533A

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG3.R68_NUM

```

44 06004001000002583679000000007601207C000000004E404EB9000000004A41660017
45 20002843000D0000025D00190000026500030000026D17
46 060040010000027200CA367900000000207C000000004E4041FAFD9610280007323C27
47 20001C43000D00000279001A0000027F5E
48 060040010000028C03004EB90000000053413401323C03004EB90000000008C0000414
49 20001C430000000002930001000002A148
50 06004001000002A60C420001660C08C0000513FC0000000000341140000743FAFD426C
51 22000C43000002B7D6
52 06004001000002C0336800000000136800020002336800040004136800060006136838
53 06004001000002DA000700071368000800081368000900091368000A000A1368000BA2
54 06004001000002F4000B1368000C000C1368000D000D1368000E000E1368000F000F60
55 060040010000030E13680010001013680011001123680012001213680016001613689F
56 0600400100000328001700171368001800181368001900196000045245FAFCC23569B7
57 060040010000034200000000156900020002356900040004156900060006156900073D
58 060040010000035C00071569000800081569000900091569000A000A1569000B000B0F
59 06004001000003761569000C000C1569000D000D1569000E000E1569000F000F15695E
60 0600400100000390001000101569001100112569001200121569001600161569001775
61 06003801000003AA0017156900180018156900190019367900000000207C54
62 20001043000D000003BDC0
63 06004001000003C0000000004E4013FC00000000003C600003BC41FAFC461028000742
64 22000C43000003CDBF
65 20001043001A000003C3AD
66 06004001000003DA323C03004EB90000000053413401323C03004EB9000000000C42D5
67 20001C430000000003E30001000003F1A6
68 06004001000003F40001660C08C0000513FC000000000034088000041140000743FA1E
69 22000C430000040388
70 060040010000040EFBF233680000000013680002000233680004000413680006000676
71 06004001000004281368000700071368000800081368000900091368000A000A1368E2
72 0600400100000442000B000B1368000C000C1368000D000D1368000E000E1368000F14
73 060040010000045C000F136800100010136800110011236800120012136800160016BC
74 060038010000047613680017001713680018001813680019001913FC000136
75 060040010000048C0000003C600002FA3679000000007601207C000000004E404EB93A
76 22000C430000048FFC
77 20001C43000D000004990019000004A119
78 06003801000004A6000000004A4166000DA4EB9000000004A41663841F9E2
79 20001C430002000004A90003000004B516
80 06004001000004BC0000000303C00065340E540207000004EB9000000004A40660048
81 20001C43000E000004BF0005000004D1D6
82 06004001000004D600B2367900000000207C000000004E4013FC0001000000376000AD
83 22000C43000004ED9E
84 20001C43000D000004DD001A000004E392
85 06004001000004F0029C45FAFB0C35690000000015690002000235690004000415699D
86 060040010000050A000600061569000700071569000800081569000900091569000A6C
87 0600400100000524000A1569000B000B1569000C000C1569000D000D1569000E000E2A
88 060040010000053E1569000F000F15690010001015690011001125690012001215696C
89 06003C01000005580016001615690017001715690018001815690019001936797B
90 060040010000057000000000207C000000004E4013FC000000000376000020645FA2D

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG3.R68_NUM

91 22000C430000058307
92 20001C43000D00000573001A0000057964
93 060040010000058AFA903569000000015690002000235690004000415690006000650
94 06004001000005A41569000700071569000800081569000900091569000A000A156956
95 06004001000005BE000B000B1569000C000C1569000D000D1569000E000E1569000F8B
96 06004001000005D8000F15690010001015690011001125690012001215690016001633
97 06003C01000005F215690017001715690018001815690019001948E700203679BE
98 060040010000060A00000000207C000000004E404CDF040041FAF9E4316A000000009D
99 20001C43000D0000060D001A000006132E
100 0600400100000624116A00020002316A00040004116A00060006116A00070007116AE2
101 060040010000063E00080008116A00090009116A000A000A116A000B000B116A000C31
102 0600400100000658000C116A000D000D116A000E000E116A000F000F116A00100010EF
103 0600400100000672116A00110011216A00120012116A00160016116A00170017116A2A
104 060040010000068C00180018116A0019001913FC000000000037102800060C000003B7
105 22000C430000069DEC
106 06004001000006A6661413FC00010000003B13FC000000000034600000D213FC0000C4
107 22001443000006AF000006B715
108 06003C01000006C00000003B0C000001663A1228000708010006661413FC000135
109 22000C43000006C3C6
110 06004001000006D80000003513FC000100000034600000A613FC00010000003A13FC03
111 22001C43000006DB000006E3000006EFC0
112 06004001000006F200010000003413FC00010000003C6000008A0C000006661A13FCB5
113 22001443000006F7000006FF85
114 060040010000070C00010000003613FC00010000003413FC00010000003A60681228DF
115 22001C430000071100000719000007211F
116 0600400100000726000713FC00000000003D0801000767460881000743FAF8DE134185
117 22000C430000072F59
118 060038010000074000074A3900000039671A13FC00010000003813FC0001DE
119 220014430000074700000751E1
120 06004001000007560000003D13FC000000000039601813FC000000000380881000788
121 22001C4300000759000007610000076B45
122 06003C01000007701141000743FAF8A41341000713FC00010000003413FC000165
123 22000C430000078305
124 06004001000007880000003C4A3900000037660001EE4EB90000000041FAF8621D7CAA
125 220014430000078B000007915D
126 2000104300040000079BE7
127 06004001000007A20012C0001D7C003200013D68000400021D68000800041D68000968
128 06004001000007BC00051D68000A00061D68000B00071D68000C00081D68000D000991
129 06004001000007D61D68000E000A1D68000F000B1D680006000C1D680007000D102E32
130 06003801000007F0000B4A79C0000000670608C000056004088000054A790E
131 200010430015000007F77A
132 06004001000008060000000670608C000046004088000041D40000B0C2E0006000CCE
133 2000104300160000080966
134 06003C010000082066061D7C0012000110280007080000056700014A2A4E4EB900
135 0600400100000838000000001D7C00110000422E00013D6D000200023D7C00020004F1
136 2000104300040000083B46
137 0600400100000852102D000D0800000666063D7C000200041D680010001C1D68001195

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG3.R68_NUM

138 060040010000086C001D2D680012001E1D68001600281D68001700271D68000800222E
139 06004001000008861D68000900231D68000A00241D68000B00251D68000C00261D68D6
140 06004001000008A0001800291D680019002A422E002C102D000D0800000667061D7A10
141 06004001000008BAF783002C2D4D0008102800060C00000167000086488045FAF77E1B
142 06004001000008D447FAF76A55401D7300000007E5402D720000000C102800060C00F5
143 06004001000008EE000367260C000002667C4E711039000000000200000C676E4E7199
144 200010430017000008FF6F
145 060040010000090813FC00010000003B13FC0000000000343B7C21120000422D0003BE
146 220014430000090F000009174F
147 0600400100000922422D0009102D000A020000E0EA0852001B4000091B6E00030002B7
148 06003C010000093C3D6D0002000241ED000A303C0004425851C8FFFC60224A396F
149 060040010000095400000035660E1D7AF6E200072D7AF6E6000C600C1D7AF6D50007D9
150 22000C43000009572F
151 060038010000096E2D7AF6DC000C4EB900000000060064EB9000000002079B8
152 20001C43000C00000979000C000009815D
153 0600380100000984000000002279000000003039000000000B15032902079D4
154 20002843000F0000098700100000098D00110000099383
155 060040010000099A000000002279000000003039000000000C3900000000003C670426
156 22000C43000009B1D5
157 2000284300120000099D0013000009A30014000009A938
158 06002C01000009B4815060044640C15032904CDF7FFF4E7516
159 04000E000100000000ED

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG5.R68_NUM

14.14 DFG5.R68_NUM

***** Source Listing —> DFG5.R68_NUM *****

```
1 02002408DFG5****000008GADMINUC000001B0014B
2 18003E08DS5A****0008RELPK**0008BCDBIN**0008SIGTIME*0008DFG5T***0082
3 18003E08SIGREQ**0008DFG5DAT10008DFG5OUT10008DFG5MSK10008DFG5DAT20011
4 18003E08DFG5OUT20008DFG5MSK20008?ACTIVE*FF08??EXIT**0008??SCHED*0083
5 18001A08??TWAIT*0008??WRITE*002E
6 160018010000001408DFG5ST**0045
7 06002C0100000000000000C000000000001000000100010A0
8 22000C43000000038C
9 06002801000000142F0A347800002C6A0058245F4239EC
10 20001003000C00000019A8
11 06004001000000220000000024790000000026790000000036390000000875236924B
12 2000344300050000002500060000002B00070000003100080000003797
13 060040010000003C0C2E0001000167624E7123FC000000080000000448E70002428398
14 220014430000004B0000004FED
15 06004001000000563679000000003E3C00022C7C00000000428699CC9BCD08C3000030
16 22000C43000000652A
17 2000104300000000005B32
18 0600400100000070207C000000004E404CDF4000247900000000267900000000363909
19 20002843001000000075000900000081000A000000087D5
20 060040010000008A000000008752369248E700027C027A01207C000000004E404CDF0F
21 20001C43000B00000008D000F0000009F3B
22 06004001000000A4400043FAFF6841EE0002700A4EB9000000000C8108F0D1806D0834
23 200010430002000000B5D6
24 06004001000000BE08E900070000600608A90007000008E90006000008A90005000038
25 06004001000000D80C813B9ACA006D0608E9000000000142E0002E18A142E0003E18AF2
26 06004001000000F2142E0004E18A142E0005E88A13420003E08A13420002E08A134285
27 060040010000010C000123FC000000100000000448E7000242833679000000003E3C59
28 2200144300000113000001175B
29 2000104300000000012369
30 060040010000012600022C7C00000000428699CC9BCD08C30000207C000000004E405E
31 22000C430000012D61
32 2000104300100000013D3F
33 06004001000001404CDF4000247900000000267900000000363900000000B752369291
34 20002843000900000149000A0000014F000B0000015567
35 06003C010000015A4EB9000000004A79000000006E2033FC00020000000036792A
36 2000284300010000015F00030000016500030000016F38
37 060040010000017200000007E047C647A0142844283207C000000004E4033FC000283
38 20001C43000400000175000E0000018573
39 060040010000018C00000000247900000000267900000000363900000000B7523692B0
40 2000344300030000018F00060000019500070000019B0008000001A1ED
41 06002001000001A64284207C000000004E4042
42 20001043000D000001ADD2
43 04000E000100000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG5T.R68_NUM

14.15 DFG5T.R68_NUM

***** Source Listing —> DFG5T.R68_NUM *****

```
1 02002408DFG5T***000008GADMINUC000000780166
2 18003E08DS5A****0008DFG5T***0008??EXIT**0008??SCHED*0008??WRITE*00D7
3 16002C010000001008SIGTIME*000000001208DFG5TST*0065
4 06001C010000000000010001000010000BC
5 0600400100000010000004790001000000106F1A36790000000007E047C647A01428440
6 22000C430000001976
7 2000104300010000000216B
8 06003C010000002A4283207C000000004E40603823FC000000000000000823FCC6
9 220014430000003B00000003F0D
10 20001043000300000003159
11 0600380100000042000000040000000C42833679000000003E3C00022C7CD7
12 2200144300000045000000049F9
13 2000104300000000000513C
14 060040010000005800000008428699CC9BCD08C30000207C000000004E404284207C6D
15 22000C4300000005B34
16 20001043000400000006B1E
17 0600180100000072000000004E40E1
18 20001043000200000007516
19 04000E0001000000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG6.R68_NUM

14.16 DFG6.R68_NUM

***** Source Listing —> DFG6.R68_NUM *****

```
1 02002408DFG6****000008GADMINUC000000E60115
2 18003E08SIGREQ**0008ACORREQ*0008ACORAV**0008CURDF***0008GETPAK**0012
3 18003E08DFG5****0008DFG8****0008DFG6DAT*0008DFG6OUT*0008DFG6MSK*009D
4 18001A08??EXIT**0008??SCHED*006B
5 160018010000000008DFG6ST**0058
6 06003C01000000002079000000002279000000003039000000000815032904A390A
7 20002843000700000005000800000000B00090000000113C
8 060040010000000180000000067684EB90000000001D7C001300004BF90000000001D6D51
9 20002843000000000001B000400000023000300000002F01
10 06004001000000032000600011D6D000800021D6D000900031D6D000A00041D6D000B29
11 0600400100000004C00051D6D000C00061D6D000E000772040C2E000100016602428150
12 0600380100000006648E780C43679000000007E022C017A01280E4283207C7A
13 20001043000500000006F19
14 0600400100000007C000000004E404CDF23014A39000000000674A4EB90000000004BF9E1
15 20002843000B0000007F000100000008B0004000000093C8
16 06004001000000096000000001D7C000100003D6D000400021D6D000600041D6D0007B4
17 20001043000300000009F1
18 06003C01000000B0000548E780C03679000000007E037C047A01280E4283207CD7
19 2000104300060000000BBCC
20 06003C01000000C80000000004E404CDF03014239000000000B15032904284207C98
21 20001C43000B0000000CB0001000000D7D3
22 060018010000000E00000000004E4073
23 20001043000A0000000E3A0
24 04000E000100000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFG8.R68_NUM

14.17 DFG8.R68_NUM

***** Source Listing —> DFG8.R68_NUM *****

```

1 02002408DFG8****000008GADMINUC0000015601A2
2 18003E08ACORAV**0008ACORREQ*0008ACBITR**0008DFPE2***0008DSI7****0076
3 18003E08DS7B****0008RELPK**0008GETPAK**0008DFG8DAT*0008DFG8OUT*0053
4 18003E08DFG8MSK*0008?ACTIVE*FF08??EXIT**0008??READW*0008??SCHED*0060
5 18000E08??WRITE*0009
6 160018010000001808DFG8ST**003E
7 06003801000000000000000800000000A000100010000001400000016000182
8 22002443000000003000000070000000F000000134B
9 0600400100000018267900000000287900000000363900000000875338932F0A347868
10 2000284300080000001D0009000000023000A000000029F1
11 06002001000000320000226A0058245F4239C5
12 20001003000B0000000338F
13 060040010000003C000000004EB9000000001D7C000000003D69000200021D690004A9
14 20001C4300010000003F0007000000045F5
15 060040010000005600041D6900050005CD494EB900000000137C0001000648E70040AD
16 20001043000600000006522
17 060040010000007042833679000000003E3C00012C7C00000000428699CC9BCD08C352
18 22000C43000000810E
19 20001043000500000007711
20 060038010000008A0000207C000000004E404CDF020048E7004042833679FD
21 20001043000F000000091ED
22 06004001000000A0000000003E3C00012C7C00000000C7C3299CC9BCD08C3000708C3D2
23 22000C43000000ADE2
24 2000104300040000000A3E6
25 06003001000000BA000F207C000000004E404CDF02002F0A3478C4
26 20001043000D0000000C1BF
27 06002401000000CC0000222A0058245F13FC0001D2
28 20001003000B9000000CDF5
29 06004001000000D8000000000C410001660C4A79000000166704422900060C29000F28
30 22000C43000000E7A8
31 200010430000000000DBB2
32 06004001000000F2000566284A39000000006606137C000200063679000000007E057C
33 20001C430000000000FB0002000001097A
34 060040010000010C4286428528094283207C000000004E4060183679000000007E0454
35 20001C43000E0000011900030000012332
36 06003C01000001264286428528094283207C000000004E402679000000002879A7
37 20001C43000E0000013300080000013BFB
38 06003C010000013E00000000363900000000B75338934284207C000000004E404A
39 20002843000900000141000A00000147000C0000015378
40 04000E000100000000ED

```

14.18 DFIN.R68 NUM

103

14.19 DFOUT.R68 NUM

104

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFPE1.R68_NUM

14.20 DFPE1.R68_NUM

***** Source Listing --> DFPE1.R68_NUM *****

```
1 02002408DFPE1***000008GADMINUC0000008C015C
2 18003E08DFG1***0008GETPAK**0008RELPK**0008?ACTIVE*FF08??EXIT**0013
3 18000E08??SCHED*002D
4 160018010000000008DFPE1E**0071
5 06002801000000002FOA34780000206A0058245F4EB980
6 200010030003000000005C5
7 0600400100000000E000000001D7C001200003D68000200021D68000400041D68000540
8 2000104300010000000117B
9 060040010000002800051D68000600061D68000700071D68000800081D68000900093C
10 06004001000000421D68000A000A1D68000B000B1D68000C000C1D68000D000D48E7D8
11 060040010000005C00803679000000007E0342864285280E4283207C000000004E40F9
12 20001C43000000000063000500000073A6
13 06003801000000764CDF01002C484EB9000000004284207C000000004E40B4
14 20001C4300020000008100040000008971
15 04000E000100000000ED
```


14.21 DFPE2.R68__NUM

106

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFPE2.R68_NUM

```

44 06004001000002000060E44E429851CEFFFC1D7C00140000422E00013D690002000269
45 060040010000021A1D690022000C1D690023000D1D690024000E1D690025000F1D693B
46 060040010000023400260010422E001108290000000167000100246900140C2A001546
47 060040010000024E000066000092302A0004C07AFDB6660000E64AAA000866184AAA6C
48 0600400100000268000C6612284E4DFAFDA24EB9000000002C4C600000C84AAA0018BC
49 200010430005000002770F
50 060040010000028266084AAA001C670000BA1D7C000100113D6A000400121D6A0006A1
51 060040010000029C00141D6A000700152D6A000800182D6A000C001C3D6A0010002017
52 06004001000002B63D6A001200223D6A001400243D6A001600262D6A001800282D6AF6
53 06004001000002D0001C002C2D6A002000302D6A0024003460624E711D7C000100119D
54 06004001000002EA1D7C00030014422E0015202A000EE1802D400018202A00124200BC
55 06004001000003042D40001C3D6A001600203D6A001800223D6A001A00243D6A001CC3
56 060040010000031E0026102A000402400003E34041FAFCD43D7000000012303C00078F
57 060040010000033841EE002830FCFFFF51C8FFFA422E003A08290001000167000100A6
58 0600400100000352266900180C2B0015000066000092302B0004C07AFCAA660000E6EE
59 060040010000036C4AAB000866184AAB000C6612284E4DFAFCBB4EB9000000002C4C63
60 2000104300050000038302
61 0600400100000386600000C84AAB001866084AAB001C670000BA1D7C0002003A3D6BDE
62 06004001000003A00004003C1D6B0006003E1D6B0007003F2D6B000800402D6B000CB8
63 06004001000003BA00443D6B001000483D6B0012004A3D6B0014004C3D6B0016004EA0
64 06004001000003D42D6B001800502D6B001C00542D6B002000582D6B0024005C6062F0
65 06004001000003EE4E711D7C0002003A1D7C0003003E422E003F202B000EE1802D4084
66 06004001000004080040202B001242002D4000443D6B001600483D6B0018004A3D6B65
67 0600400100000422001A004C3D6B001C004E102B000402400003E34041FAFBC83D70C9
68 060040010000043C0000003C303C000741EE005030FCFFFF51C8FFFA4A2E0011660C14
69 06004001000004564A2E003A66061D7C0001000148E700403679000000007E01428641
70 2000104300000000046B1E
71 06003801000004704285280E4283207C000000004E404CDF02002C494EB9B8
72 2000104300090000047B05
73 0600400100000486000000004A3900000000670C4239000000004EB9000000004284F1
74 2000344300020000048900040000048F00040000049700030000049D00
75 06001C01000004A0207C000000004E400F
76 200010430008000004A5DC
77 04000E000100000000ED

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFPON.R68_NUM

14.22 DFPON.R68_NUM

```
***** Source Listing —> DFPON.R68_NUM *****  
*****  
1 02002408DFPON***000008ADFUC***00000070019B  
2 18003208?DVRCB**0008INITDF**0008DFDIB***0008ACUINIT*00B5  
3 160018010000000008DFPON***0060  
4 060040010000000048E7FFFE367900000000424542472C7C000000004286387C0000AA  
5 20001043000200000000982  
6 060040010000001A2A7C0000000042834EB9000000003279000000004252328A31FC05  
7 20001C4300000000002700020000002D2B  
8 06004001000000340040F80C31FCA000F4C231FCEFFFF4C231FC2000F4C231FC6FFF4F  
9 06003C010000004EF4C231FC114DF80008380000F81367F84EB9000000004EB97E  
10 2000104300030000006327  
11 0600200100000066000000004CDF7FFF4E7507  
12 2000104300010000006923  
13 04000E000100000000ED
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFRPT.R68_NUM

14.23 DFRPT.R68_NUM

***** Source Listing —> DFRPT.R68_NUM *****

```

1 02002408DFRPT***000008GADMINUC000004860135
2 18003E08RELPAK**0008GETBITS*0008HEXSTR**0008WRITEH**0008GETPRMPT0083
3 18002608?ACTIVE*FF08??EXIT**0008??SCHED*00BA
4 160040010000007208TFLAG1**000000007408TFLAG2**000000005908QUIT****0022
5 16002C010000007608REPIP***000000007808DFRPTST*00A0
6 06001C01000000000000000100000002DA
7 06004001000000580000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF79
8 06004001000000720000000000002F0A347800002A6A0058245F41FAFFD40C2D004C60
9 2000100300050000007D4B
10 060040010000008C0000671041FAFFD00C2D004D0000670441FAFFCC102D0001323C09
11 06004001000000A603044EB9000000000C410001670258884AA800006A06214D00009E
12 200010430001000000ADDF
13 06004001000000C0600E2C6800004EB900000000214D00004A790000007267464E71E1
14 22000C43000000D5BA
15 200010430000000000CBC2
16 06004001000000DA4AB90000005A6B3C4E71207AFF74302800024AB9000000626B2ABB
17 22001443000000DF000000F1B7
18 06004001000000F44E71207AFF6AB0680002661E4E714AB90000006A6B144E71207A61
19 22000C430000010787
20 060040010000010EFF5CB068000266084E714246604E4E714A7900000074670003561C
21 22000C43000001236B
22 06004001000001284AB90000005E6B00034C207AFF2A302800024AB9000000666B0084
23 220014430000012D0000013F19
24 0600400100000142033A207AFF20B06800026600032E4AB90000006E6B000324207A32
25 22000C430000015539
26 060040010000015CFF12B0680002660003183C3C000433FC00010000007647FAFEE669
27 22000C43000001711D
28 06004001000001762073600013FC00000000003C13FC000D0000003B43FAFEAF342867
29 2200144300000181000001897B
30 06004001000001900016323C00044EB900000000133C002034280014323C00044EB941
31 2000104300020000019BEF
32 06004001000001AA00000000133C002034280012323C00044EB900000000133C002049
33 20001C430002000001AD0002000001BF0F
34 06004001000001C434280010323C00044EB900000000133C00202428000C323C0008D2
35 200010430002000001D1B9
36 06004001000001DE4EB900000000133C002024280008323C00084EB900000000133C44
37 20001C430002000001E30002000001F5A3
38 06004001000001F8002014280007323C00024EB900000000133C002014280006323CC7
39 2000104300020000020782
40 060038010000021200024EB900000000133C002034280004323C00044EB95C
41 2000104300020000021970
42 060038010000022800000000133C002045FAFDCE24326000323C00014EB9F2
43 2000104300020000022B5E

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFRPT.R68_NUM

```

44 060040010000023E00000000133C00204DFAFDC04EB90000000004EB90000000013FCE9
45 2000284300020000024100030000024F00040000025581
46 060040010000025800000000003113FC000D0000003043FAFDC8363C0008E34345E813
47 220014430000025D00000265C1
48 06004001000002720018D4C3363C00085343323C000434224EB900000000133C002048
49 2000104300020000028702
50 06003C010000028C51CBFFF24DFAFD764EB90000000004EB9000000002C484EB9DF
51 20001C4300030000029900040000029F3E
52 06003C01000002A40000000027BCFFFFFFFFF600047FAFDB02073600013FC0000E8
53 200010430000000002A7E4
54 06004001000002BC0000003613FC000D0000003543FAFD6B363C0005E54345E80004FF
55 22001443000002BF000002C7FD
56 06004001000002D6D4C3363C00055343323C000824224EB900000000133C002051CBEF
57 200010430002000002E9A0
58 06003801000002F0FFF24DFAFD144EB90000000004EB90000000013FC000069
59 20001C430003000002FB00040000030179
60 06004001000003060000002D13FC000D0000002C43FAFD18363C0009E54345E8000415
61 22001443000003090000031167
62 0600400100000320D4C3363C00045343323C000824224EB900000000133C002051CBA5
63 2000104300020000033355
64 060038010000033AFF24DFAFCCA4EB90000000004EB9000000002C484EB9FD
65 20001C4300030000034500040000034BE4
66 06003C01000003500000000027BCFFFFFFFFF600047FAFD0C2073600013FC0000DF
67 2000104300000000035337
68 06004001000003680000002D13FC000D0000002C43FAFCB6363C0004E54345E800041B
69 220014430000036B00000373A3
70 0600400100000382D4C3363C00045343323C000824224EB900000000133C002051CB43
71 20001043000200000395F3
72 060038010000039CFFF24DFAFC684EB90000000004EB90000000013FC000069
73 20001C430003000003A70004000003AD20
74 06004001000003B20000002D13FC000D0000002C43FAFC6C34280022323C00044EB9F3
75 22001443000003B5000003BD0F
76 06004001000003CC00000000133C002034280020323C00044EB900000000133C002017
77 20001C430002000003CF0002000003E1C7
78 06004001000003E63428001E323C00044EB900000000133C00203428001C323C000484
79 200010430002000003F395
80 06004001000004004EB900000000133C00203428001A323C00044EB900000000133C01
81 20001C4300020000040500020000041759
82 060038010000041A002014280018323C00024EB900000000142800174EB95E
83 200010430002000004295E
84 060040010000043000000000142800164EB900000000142800154EB900000000142898
85 2000284300020000043300020000043D000200000447AC
86 06003C010000044A00144EB900000000133C00204DFAFB04EB90000000004EB9E5
87 20001C4300020000045100030000045FC4
88 060038010000046200000002C484EB90000000027BCFFFFFFFFF60004279E6
89 20001C4300040000046500000000046DA3
90 0600280100000478000000764284207C0000000004E40EF

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFRPT.R68_NUM

91 22000C430000047B10
92 2000104300060000048300
93 04000E0001000000000ED

Improved GUARDRAIL V MC68000 'DF' Files

15 FILES WITH EXTENSION '.DEF'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

15.1 DFSMENU1.DEF_NUM

```

***** Source Listing --> DFSMENU1.DEF_NUM *****
*****
1  DS.W 0 ;word align menu
2  DFSMENU1 DS.W 0 ;label of menu
3  DC.B $1B,$4B,$1B,$4A ;asc H esc J
4  * FDF SIO TEST FOR DPU UNDER TEST
5  DC.B $20,$20,$46,$44,$46,$20,$53,$49,$4F,$20
6  DC.B $54,$45,$53,$54,$20,$46,$4F,$52,$20,$44
7  DC.B $50,$55,$20,$55,$4E,$44,$45,$52,$20,$54
8  DC.B $45,$53,$54
9  DC.B $0A,$0D ; lf cr
10 DC.B $0A,$0D ; lf cr
11 * THIS TEST WILL LOAD A ROUTINE FROM THE TEST SET EPROM TO
12 DC.B $20,$20,$54,$4B,$49,$53,$20,$54,$45,$53
13 DC.B $54,$20,$57,$49,$4C,$20,$4C,$4F,$41
14 DC.B $44,$20,$41,$20,$52,$4F,$55,$54,$49,$4E
15 DC.B $45,$20,$46,$52,$4F,$4D,$20,$54,$4B,$45
16 DC.B $20,$54,$45,$53,$54,$20,$53,$45,$54,$20
17 DC.B $45,$50,$52,$4F,$4D,$20,$54,$4F
18 DC.B $0A,$0D ; lf cr
19 * THE RAM OF CPU 5 IN THE DPU UNDER TEST. THAT ROUTINE WILL
20 DC.B $20,$20,$54,$4B,$45,$20,$52,$41,$4D,$20
21 DC.B $4F,$46,$20,$43,$50,$55,$20,$35,$20,$49
22 DC.B $4E,$20,$54,$4B,$45,$20,$44,$50,$55,$20
23 DC.B $55,$4E,$44,$45,$52,$20,$54,$45,$53,$54
24 DC.B $2E,$20,$20,$54,$4B,$41,$54,$20,$20,$52
25 DC.B $4F,$55,$54,$49,$4E,$45,$20,$57,$49,$4C
26 DC.B $0A,$0D
27 DC.B $0A,$0D ; lf cr
28 * PERFORM THE SIO TEST ON THE FDF SIO CARD. THIS TEST
29 DC.B $20,$20,$50,$45,$52,$46,$4F,$52,$4D,$20
30 DC.B $54,$4B,$45,$20,$53,$49,$4F,$20,$54,$45
31 DC.B $53,$54,$20,$4F,$4E,$20,$54,$4B,$45,$20
32 DC.B $46,$44,$46,$20,$53,$49,$4F,$20,$43,$41
33 DC.B $52,$44,$2E,$20,$20,$54,$4B,$49,$53,$20
34 DC.B $54,$45,$53,$54
35 DC.B $0A,$0D ; lf cr
36 * CONSISTS OF THE FOLLOWING SUB-TESTS:
37 DC.B $20,$20,$43,$4F,$4E,$53,$49,$53,$54,$53
38 DC.B $20,$4F,$46,$20,$54,$4B,$45,$20,$46,$4F
39 DC.B $4C,$4C,$4F,$57,$49,$4E,$47,$20,$53,$55
40 DC.B $42,$2D,$54,$45,$53,$54,$53,$3A
41 DC.B $0A,$0D ; lf cr
42 * 1. SIO CHANNEL A INTERRUPT TEST,
43 DC.B $20,$20,$20,$20,$20,$20,$20,$31,$2E,$20

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU1.DEF_NUM

```

44 DC.B $20,$53,$49,$4F,$20,$43,$48,$41,$4E,$4E
45 DC.B $45,$4C,$20,$41,$20,$49,$4E,$54,$45,$52
46 DC.B $52,$55,$50,$54,$20,$54,$45,$53,$54,$2C
47 DC.B $0A,$0D ; lf cr
48 *
49 2. SIO CHANNEL B INTERRUPT TEST,
50 DC.B $20,$20,$20,$20,$20,$20,$20,$32,$2E,$20
51 DC.B $20,$53,$49,$4F,$20,$43,$48,$41,$4E,$4E
52 DC.B $45,$4C,$20,$42,$20,$49,$4E,$54,$45,$52
53 DC.B $52,$55,$50,$54,$20,$54,$45,$53,$54,$2C
54 DC.B $0A,$0D ; lf cr
55 *
56 3. SIO CHANNEL C INTERRUPT TEST,
57 DC.B $20,$20,$20,$20,$20,$20,$20,$33,$2E,$20
58 DC.B $20,$53,$49,$4F,$20,$43,$48,$41,$4E,$4E
59 DC.B $45,$4C,$20,$43,$20,$49,$4E,$54,$45,$52
60 DC.B $52,$55,$50,$54,$20,$54,$45,$53,$54,$2C
61 DC.B $0A,$0D ; lf cr
62 *
63 4. SIO CHANNEL A INTERNAL LOOP TEST,
64 DC.B $20,$20,$20,$20,$20,$20,$20,$34,$2E,$20
65 DC.B $20,$53,$49,$4F,$20,$43,$48,$41,$4E,$4E
66 DC.B $45,$4C,$20,$41,$20,$49,$4E,$54,$45,$52
67 DC.B $4E,$41,$4C,$20,$4C,$4F,$4F,$50,$20,$54
68 DC.B $45,$53,$54,$2C
69 DC.B $0A,$0D ; lf cr
70 *
71 5. SIO CHANNEL B INTERNAL LOOP TEST AND
72 DC.B $20,$20,$20,$20,$20,$20,$20,$35,$2E,$20
73 DC.B $20,$53,$49,$4F,$20,$43,$48,$41,$4E,$4E
74 DC.B $45,$4C,$20,$42,$20,$49,$4E,$54,$45,$52
75 DC.B $4E,$41,$4C,$20,$4C,$4F,$4F,$50,$20,$54
76 DC.B $45,$53,$54,$20,$41,$4E,$44
77 DC.B $0A,$0D ; lf cr
78 *
79 6. SIO EXTERNAL MONITOR TEST,
80 DC.B $20,$20,$20,$20,$20,$20,$20,$36,$2E,$20
81 DC.B $20,$53,$49,$4F,$20,$45,$58,$54,$45,$52
82 DC.B $4E,$41,$4C,$20,$4D,$4F,$4E,$49,$54,$4F
83 DC.B $52,$20,$54,$45,$53,$54,$2E
84 DC.B $0A,$0D ; lf cr
85 *
86 THE TEK 318S1 LOGIC ANALYZER WILL BE USED TO TEST THE FDP SIO.
87 DC.B $20,$20,$54,$48,$45,$20,$46,$44,$46,$20
88 DC.B $53,$49,$4F,$2E
89 DC.B $0A,$0D ; lf cr
90 *
91 CONFIGURE THE TEK 318S1 USING NVM SETUP 3 (SEE SECTION 9 OF
92 DC.B $20,$20,$43,$4F,$4E,$46,$49,$47,$55,$52

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DFJDFSMENU1.DEF_NUM

91	DC.B	\$45,\$20,\$54,\$48,\$45,\$20,\$54,\$45,\$48,\$20
92	DC.B	\$33,\$31,\$38,\$53,\$31,\$20,\$55,\$53,\$49,\$4E
93	DC.B	\$47,\$20,\$4E,\$56,\$4D,\$20,\$53,\$45,\$54,\$55
94	DC.B	\$50,\$20,\$33,\$20,\$28,\$53,\$45,\$45,\$20,\$53
95	DC.B	\$45,\$43,\$54,\$49,\$4F,\$4E,\$20,\$39,\$20,\$4F
96	DC.B	\$46
97	DC.B	\$0A,\$0D ; lf cr
98	*	"318/338 OPERATOR'S MANUAL" FOR A DESCRIPTION OF NVM USAGE).
99	DC.B	\$20,\$20,\$22,\$33,\$31,\$38,\$2F,\$33,\$33,\$38
100	DC.B	\$20,\$4F,\$50,\$45,\$52,\$41,\$54,\$4F,\$52,\$27
101	DC.B	\$53,\$20,\$4D,\$41,\$4E,\$55,\$41,\$4C,\$22,\$20
102	DC.B	\$46,\$4F,\$52,\$20,\$41,\$20,\$44,\$45,\$53,\$43
103	DC.B	\$52,\$49,\$50,\$54,\$49,\$4F,\$4E,\$20,\$4F,\$46
104	DC.B	\$20,\$4E,\$56,\$4D,\$20,\$55,\$53,\$41,\$47,\$45
105	DC.B	\$29,\$2E
106	DC.B	\$0A,\$0D ; lf cr
107	DC.B	\$0A,\$0D ; lf cr
108	*	CONNECT THE FOLLOWING CABLES BEFORE PROCEEDING
109	DC.B	\$20,\$20,\$43,\$4F,\$4E,\$4E,\$45,\$43,\$54,\$20
110	DC.B	\$54,\$48,\$45,\$20,\$46,\$4F,\$4C,\$4C,\$4F,\$57
111	DC.B	\$49,\$4E,\$47,\$20,\$43,\$41,\$42,\$4C,\$45,\$53
112	DC.B	\$20,\$42,\$45,\$46,\$4F,\$52,\$45,\$20,\$50,\$52
113	DC.B	\$4F,\$43,\$45,\$45,\$44,\$49,\$4E,\$47
114	DC.B	\$0A,\$0D ; lf cr
115	DC.B	\$0A,\$0D ; lf cr
116	*	CABLE TO FROM CABLE P/N
117	DC.B	\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$43,\$41,\$42
118	DC.B	\$4C,\$45,\$20,\$20,\$54,\$4F,\$20,\$20,\$20,\$20
119	DC.B	\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20
120	DC.B	\$20,\$20,\$20,\$20,\$20,\$20,\$46,\$52,\$4F,\$4D,\$20
121	DC.B	\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20
122	DC.B	\$20,\$20,\$20,\$20,\$20,\$20,\$43,\$41,\$42,\$4C,\$45
123	DC.B	\$20,\$50,\$2F,\$4E
124	DC.B	\$0A,\$0D ; lf cr
125	*	W16 ADFU P2>J12 TEK 318S1 P1>A 10-165316-1
126	DC.B	\$20,\$20,\$20,\$20,\$20,\$20,\$57,\$31,\$36
127	DC.B	\$20,\$20,\$20,\$20,\$41,\$44,\$50,\$55,\$20,\$20
128	DC.B	\$20,\$20,\$50,\$32,\$3E,\$4A,\$31,\$32,\$20,\$20
129	DC.B	\$20,\$20,\$20,\$20,\$20,\$20,\$54,\$45,\$4B,\$20,\$33
130	DC.B	\$31,\$38,\$53,\$31,\$20,\$20,\$20,\$50,\$31
131	DC.B	\$3E,\$41,\$20,\$31,\$30,\$20,\$31,\$36,\$35
132	DC.B	\$33,\$31,\$36,\$2D,\$31
133	DC.B	\$0A,\$0D ; lf cr
134	DC.B	\$0A,\$0D ; lf cr
135	*	HIT ANY KEY TO PROCEED
136	DC.B	\$20,\$20,\$48,\$49,\$54,\$20,\$41,\$4E,\$59,\$20
137	DC.B	\$4B,\$45,\$59,\$20,\$54,\$4F,\$20,\$50,\$52,\$4F

Improved GUARDRAIL V MC68000 'DF' Files
DRCL:[ALGO.IGR.MC68000_DF]DFSMENU1.DEF_NUM

138 DC.B S43,S45,S45,S44
139 *-> DFSMENU2 BEGIN
140 DC.B S00 ; end string

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU2.DEF_NUM

15.2 DFSMENU2.DEF_NUM

```

***** Source Listing --> DFSMENU2.DEF_NUM *****
*****
1 DS.W 0 ;word align menu
2 DFSMENU2 DS.W 0 ;label of menu
3 DC.B $0A,$0D ; if cr
4 DC.B $0A,$0D ; if cr
5 DC.B $0A,$0D ; if cr
6 * THE FDF SIO DIFFERS FROM THE OTHER SIO CARDS SINCE IT
7 DC.B $20,$20,$54,$48,$45,$20,$46,$44,$46,$20
8 DC.B $53,$49,$4F,$20,$44,$49,$46,$46,$45,$52
9 DC.B $53,$20,$46,$52,$4F,$4D,$20,$54,$48,$45
10 DC.B $20,$4F,$54,$48,$45,$52,$20,$53,$49,$4F
11 DC.B $20,$43,$41,$52,$44,$53,$20,$53,$49,$4E
12 DC.B $43,$45,$20,$49,$54
13 DC.B $0A,$0D ; if cr
14 * DOES NOT USE THE LOAD STROBE GENERATED ON THE SIO CARD
15 DC.B $20,$20,$44,$4F,$45,$53,$20,$4E,$4F,$54
16 DC.B $20,$55,$53,$45,$20,$54,$48,$45,$20,$4C
17 DC.B $4F,$41,$44,$20,$53,$54,$52,$4F,$42,$45
18 DC.B $20,$47,$45,$4E,$45,$52,$41,$54,$45,$44
19 DC.B $20,$4F,$4E,$20,$54,$48,$45,$20,$53,$49
20 DC.B $4F,$20,$43,$41,$52,$44
21 DC.B $0A,$0D ; if cr
22 * BUT RATHER USES THE LOAD STROBES (ONE FOR CHANNEL A AND
23 DC.B $20,$20,$42,$55,$54,$20,$52,$41,$54,$48
24 DC.B $45,$52,$20,$55,$53,$45,$53,$20,$54,$48
25 DC.B $45,$20,$4C,$4F,$41,$44,$20,$53,$54,$52
26 DC.B $4F,$42,$45,$53,$20,$28,$4F,$4E,$45,$20
27 DC.B $46,$4F,$52,$20,$43,$48,$41,$4E,$45
28 DC.B $4C,$20,$41,$20,$41,$4E,$44
29 DC.B $0A,$0D ; if cr
30 * ONE FOR CHANNEL B) PROVIDED BY THE FDF CONTROLLER CARD.
31 DC.B $20,$20,$4F,$4E,$45,$20,$46,$4F,$52,$20
32 DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$42,$29
33 DC.B $20,$50,$52,$4F,$56,$49,$44,$45,$44,$20
34 DC.B $42,$59,$20,$54,$48,$45,$20,$46,$44,$46
35 DC.B $20,$43,$4F,$4E,$54,$52,$4F,$4C,$4C,$45
36 DC.B $52,$20,$43,$41,$52,$44,$2E
37 DC.B $0A,$0D ; if cr
38 * THE FDF CONTROLLER CARD MONITORS THE BIT CLOCK OF THE
39 DC.B $20,$20,$54,$48,$45,$20,$46,$44,$46,$20
40 DC.B $43,$4F,$4E,$54,$52,$4F,$4C,$4C,$45,$52
41 DC.B $20,$43,$41,$52,$44,$20,$4D,$4F,$4E,$49
42 DC.B $54,$4F,$52,$53,$20,$54,$48,$45,$20,$42
43 DC.B $49,$54,$20,$43,$4C,$4F,$43,$4B,$20,$4F

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU2.DEF_NUM

```

44 DC.B $46,$20,$54,$48,$45
45 DC.B SOA,$0D ; lf cr
46 * FDF SIO TO DETERMINE WHEN TO PROCEED TO ITS NEXT STATE.
47 DC.B $20,$20,$46,$44,$46,$20,$53,$49,$4F,$20
48 DC.B $54,$4F,$20,$44,$45,$54,$52,$4D,$49
49 DC.B $4E,$45,$20,$57,$48,$45,$4E,$20,$54,$4F
50 DC.B $20,$50,$52,$4F,$43,$45,$44,$20,$54
51 DC.B $4F,$20,$49,$54,$53,$20,$4E,$45,$58,$54
52 DC.B $20,$53,$54,$41,$54,$45,$52E
53 DC.B SOA,$0D ; lf cr
54 * IF THE FDF CONTROLLER DOES NOT PROVIDE THE LOAD STROBE TO
55 DC.B $20,$20,$49,$46,$20,$54,$48,$45,$20,$46
56 DC.B $44,$46,$20,$43,$4F,$4E,$54,$52,$4F,$4C
57 DC.B $4C,$45,$52,$20,$44,$4F,$45,$53,$20,$4E
58 DC.B $4F,$54,$20,$50,$52,$4F,$56,$49,$44,$45
59 DC.B $20,$54,$48,$45,$20,$4C,$4F,$41,$44,$20
60 DC.B $53,$54,$52,$4F,$42,$45,$20,$54,$4F
61 DC.B SOA,$0D ; lf cr
62 * THE FDF SIO THEN THE SIO WILL NOT INITIATE A TRANSMISSION
63 DC.B $20,$20,$54,$48,$45,$20,$46,$44,$46,$20
64 DC.B $53,$49,$4F,$20,$54,$48,$45,$4E,$20,$54
65 DC.B $48,$45,$20,$53,$49,$4F,$20,$57,$49,$4C
66 DC.B $4C,$20,$4E,$4F,$54,$20,$49,$4E,$49,$54
67 DC.B $49,$41,$54,$45,$20,$41,$20,$54,$52,$41
68 DC.B $4E,$53,$4D,$49,$53,$53,$48,$4F,$4E
69 DC.B SOA,$0D ; lf cr
70 * OF DATA. IF THE FDF SIO DOES NOT TRANSMIT DATA WHEN THE
71 DC.B $20,$20,$4F,$46,$20,$44,$41,$54,$41,$2E
72 DC.B $20,$20,$49,$46,$20,$54,$48,$45,$20,$46
73 DC.B $44,$46,$20,$53,$49,$4F,$20,$44,$4F,$45
74 DC.B $53,$20,$4E,$4F,$54,$20,$54,$52,$41,$4E
75 DC.B $53,$4D,$49,$54,$20,$44,$41,$54,$41,$20
76 DC.B $57,$48,$45,$4E,$20,$54,$48,$45
77 DC.B SOA,$0D ; lf cr
78 * FDF CONTROLLER PROVIDES THE CORRECT LOAD STROBES THEN THE
79 DC.B $20,$20,$46,$44,$46,$20,$43,$4F,$4E,$54
80 DC.B $52,$4F,$4C,$4C,$45,$52,$20,$50,$52,$4F
81 DC.B $56,$49,$44,$45,$53,$20,$54,$48,$45,$20
82 DC.B $43,$4F,$52,$52,$45,$43,$54,$20,$4C,$4F
83 DC.B $41,$44,$20,$53,$54,$52,$4F,$42,$45,$53
84 DC.B $20,$54,$48,$45,$4E,$20,$54,$48,$45
85 DC.B SOA,$0D ; lf cr
86 * FDF CONTROLLER'S STATE MACHINE WILL BE IN A STATE OF WAITING
87 DC.B $20,$20,$46,$44,$46,$20,$43,$4F,$4E,$54
88 DC.B $52,$4F,$4C,$4C,$45,$52,$27,$53,$20,$53
89 DC.B $54,$41,$54,$45,$20,$4D,$41,$43,$48,$49
90 DC.B $4E,$45,$20,$57,$49,$4C,$4C,$20,$42,$45

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU2.DEF_NUM

```

91      DC.B $20,$49,$4E,$20,$41,$20,$53,$54,$41,$54
92      DC.B $45,$20,$4F,$46,$20,$57,$41,$49,$54,$49
93      DC.B $4E,$47
94      DC.B $0A,$0D      ; lf cr
95  * FOR FDF SIO CLOCK PULSES WHICH WILL NEVER APPEAR. IF THE
96      DC.B $20,$20,$46,$4F,$52,$20,$46,$44,$46,$20
97      DC.B $53,$49,$4F,$20,$43,$4C,$4F,$43,$4B,$20
98      DC.B $50,$55,$4C,$53,$45,$53,$20,$57,$48,$49
99      DC.B $43,$48,$20,$57,$49,$4C,$4C,$20,$4E,$45
100     DC.B $56,$45,$52,$20,$41,$50,$50,$45,$41,$52
101     DC.B $2E,$20,$20,$49,$46,$20,$54,$48,$45
102     DC.B $0A,$0D      ; lf cr
103  * FDF SIO DOES NOT APPEAR TO BE FUNCTIONING IT WILL HAVE TO
104     DC.B $20,$20,$46,$44,$46,$20,$53,$49,$4F,$20
105     DC.B $44,$4F,$45,$53,$20,$4E,$4F,$54,$20,$41
106     DC.B $50,$50,$45,$41,$52,$20,$54,$4F,$20,$42
107     DC.B $45,$20,$46,$55,$4E,$43,$54,$49,$4F,$4E
108     DC.B $49,$4E,$47,$20,$49,$54,$20,$57,$49,$4C
109     DC.B $4C,$20,$48,$41,$56,$45,$20,$54,$4F
110     DC.B $0A,$0D      ; lf cr
111  * BE DETERMINED IF THE FDF CONTROLLER IS SUPPLYING THE CORRECT
112     DC.B $20,$20,$42,$45,$20,$44,$45,$54,$45,$52
113     DC.B $4D,$49,$4E,$45,$44,$20,$49,$46,$20,$54
114     DC.B $48,$45,$20,$46,$44,$46,$20,$43,$4F,$4E
115     DC.B $54,$52,$4F,$4C,$4C,$45,$52,$20,$49,$53
116     DC.B $20,$53,$55,$50,$50,$4C,$59,$49,$4E,$47
117     DC.B $20,$54,$48,$45,$20,$43,$4F,$52,$52,$45
118     DC.B $43,$54
119     DC.B $0A,$0D      ; lf cr
120  * LOAD STROBES BEFORE STATING THAT THE FDF SIO IS FAULTY.
121     DC.B $20,$20,$4C,$4F,$41,$44,$20,$53,$54,$52
122     DC.B $4F,$42,$45,$53,$20,$42,$45,$46,$4F,$52
123     DC.B $45,$20,$53,$54,$41,$54,$49,$4E,$47,$20
124     DC.B $54,$48,$41,$54,$20,$54,$48,$45,$20,$46
125     DC.B $44,$46,$20,$53,$49,$4F,$20,$49,$53,$20
126     DC.B $46,$41,$55,$4C,$54,$59,$52E
127     DC.B $0A,$0D      ; lf cr
128     DC.B $0A,$0D      ; lf cr
129  * HIT ANY KEY TO PROCEED
130     DC.B $20,$20,$48,$49,$54,$20,$41,$4E,$59,$20
131     DC.B $48,$45,$58,$20,$54,$4F,$20,$50,$52,$4F
132     DC.B $43,$45,$44
133  *-> DFSMENU3 OR DFSMENU4 OR BUSERR BEGIN
134     DC.B $00      ; end string

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU3.DEF_NUM

15.3 DFSMENU3.DEF_NUM

```
***** Source Listing -> DFSMENU3.DEF_NUM *****
*****
1  DS W 0 ;word align menu
2  DFSMENU3 DS W 0 ;label of menu
3  DC B SOA,$0D ; lf cr
4  DC B SOA,$0D ; lf cr
5  * TEST SET DPU NOT ABLE TO READ RESULT OF FDF SIO TEST;
6  DC B $20,$20,$54,$45,$53,$54,$20,$53,$54,$20,$53,$45,$54
7  DC B $20,$44,$50,$55,$20,$4E,$4F,$54,$20,$54,$1
8  DC B $42,$4C,$45,$20,$54,$4F,$20,$52,$45,$41
9  DC B $44,$20,$52,$45,$53,$55,$4C,$54,$20,$4F
10 DC B $46,$20,$46,$44,$46,$20,$53,$49,$4F,$20
11 DC B $54,$45,$53,$54,$3B
12 DC B SOA,$0D ; lf cr
13 * FDF SIO CARD (10-156022-1, A13) ASSUMED FAULTY.
14 DC B $20,$20,$46,$44,$46,$20,$53,$49,$4F,$20
15 DC B $43,$41,$52,$44,$20,$28,$31,$30,$2D,$31
16 DC B $35,$36,$30,$32,$32,$2D,$31,$2C,$20,$41
17 DC B $31,$33,$29,$20,$41,$53,$53,$55,$4D,$45
18 DC B $44,$20,$46,$41,$55,$4C,$54,$59,$2E
19 DC B SOA,$0D ; lf cr
20 DC B SOA,$0D ; lf cr
21 * HIT "Q" TO QUIT
22 DC B $20,$20,$48,$49,$54,$20,$22,$51,$22,$20
23 DC B $54,$4F,$20,$51,$55,$49,$54
24 DC B SOA,$0D ; lf cr
25 * OR "R" TO REPEAT
26 DC B $20,$20,$4F,$52,$52,$20,$52,$22,$20
27 DC B $54,$4F,$20,$52,$45,$50,$45,$41,$54
28 DC B $0D ; end string
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU4.DEF_NUM

15.4 DFSMENU4.DEF_NUM

```
***** Source Listing --> DFSMENU4.DEF_NUM *****
*****
1  DS.W 0 ;word align menu
2  DFSMENU4 DS.W 0 ;label of menu
3  DC.B SOA,$0D ; lf cr
4  DC.B SOA,$0D ; lf cr
5  * RESULTS OF PDF SIO TEST
6  DC.B $20,$20,$52,$45,$53,$55,$4C,$54,$54,$53,$20
7  DC.B $4F,$46,$20,$46,$44,$46,$20,$53,$49,$4F
8  DC.B $20,$54,$45,$53,$54
9  DC.B SOA,$0D ; lf cr
10 * CHANNEL A INTERRUPT TEST
11 DC.B $20,$20,$20,$20,$43,$48,$41,$4E,$4E,$45
12 DC.B $4C,$20,$41,$20,$49,$4E,$54,$45,$52,$52
13 DC.B $55,$50,$54,$20,$54,$45,$53,$54,$20,$20
14 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$3E
15 DC.B $00 ; end string
16 DFSLINE1 DS.W 0 ;label of sub menu
17 DC.B SOA,$0D ; lf cr
18 * CHANNEL B INTERRUPT TEST
19 DC.B $20,$20,$20,$20,$43,$48,$41,$4E,$4E,$45
20 DC.B $4C,$20,$42,$20,$49,$4E,$54,$45,$52,$52
21 DC.B $55,$50,$54,$20,$54,$45,$53,$54,$20,$20
22 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$3E
23 DC.B $00 ; end string
24 DFSLINE2 DS.W 0 ;label of sub menu
25 DC.B SOA,$0D ; lf cr
26 * CHANNEL C INTERRUPT TEST
27 DC.B $20,$20,$20,$20,$43,$48,$41,$4E,$4E,$45
28 DC.B $4C,$20,$43,$20,$49,$4E,$54,$45,$52,$52
29 DC.B $55,$50,$54,$20,$54,$45,$53,$54,$20,$20
30 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$3E
31 DC.B $00 ; end string
32 DFSLINE3 DS.W 0 ;label of sub menu
33 DC.B SOA,$0D ; lf cr
34 * CHANNEL A INTERNAL LOOP TEST
35 DC.B $20,$20,$20,$20,$43,$48,$41,$4E,$4E,$45
36 DC.B $4C,$20,$41,$20,$49,$4E,$54,$45,$52,$4E
37 DC.B $41,$4C,$20,$4C,$4F,$4F,$50,$20,$54,$45
38 DC.B $53,$54,$20,$20,$20,$20,$20,$20,$3E
39 DC.B $00 ; end string
40 DFSLINE4 DS.W 0 ;label of sub menu
41 DC.B SOA,$0D ; lf cr
42 * CHANNEL B INTERNAL LOOP TEST
43 DC.B $20,$20,$20,$20,$43,$48,$41,$4E,$4E,$45
```


Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFSMENU4.DEF_NUM

```

44 DC B $4C,$20,$42,$20,$49,$4E,$54,$45,$52,$4E
45 DC B $41,$4C,$20,$4C,$4F,$50,$20,$54,$45
46 DC B $53,$54,$20,$20,$20,$20,$20,$20,$3E
47 DC B $00 ; end string
48 DFSLINES DS W 0 ;label of sub menu
49 DC B $0A,$0D ; lf cr
50 DC B $0A,$0D ; lf cr
51 * EXTERNAL MONITOR TEST IS CURRENTLY RUNNING
52 DC B $20,$20,$20,$20,$45,$58,$54,$45,$52,$4E
53 DC B $41,$4C,$20,$4D,$4F,$4E,$49,$54,$4F,$52
54 DC B $20,$54,$45,$53,$54,$20,$49,$53,$20,$43
55 DC B $55,$52,$52,$45,$4E,$54,$4C,$59,$20,$52
56 DC B $55,$4E,$4E,$49,$4F,$47
57 DC B $0A,$0D ; lf cr
58 DC B $0A,$0D ; lf cr
59 * HIT ANY KEY TO PROCEED
60 DC B $20,$20,$48,$49,$54,$20,$41,$4E,$59,$20
61 DC B $48,$45,$59,$20,$54,$4F,$20,$50,$52,$4F
62 DC B $43,$45,$45,$44
63 *-> DFSMENU5 BEGIN
64 DC B $00 ; end string

```

15.5 DFSMENU5.DEF_NUM

```

***** Source Listing --> DFSMENU5.DEF_NUM *****
*****
1      DS.W 0      ;word align menu
2  DFSMENU5 DS.W 0      ;label of menu
3      DC.B $0A,$0D      ; lf cr
4      DC.B $0A,$0D      ; lf cr
5  * FDF SIO MONITOR TEST RESULTS
6      DC.B $20,$20,$46,$44,$46,$20,$53,$49,$4F,$20
7      DC.B $4D,$4F,$4E,$49,$54,$4F,$52,$20,$54,$45
8      DC.B $53,$54,$20,$52,$45,$53,$55,$4C,$54,$54,$53
9      DC.B $0A,$0D      ; lf cr
10     DC.B $0A,$0D      ; lf cr
11  * THE BIT PATTERN SENT FROM THE FDF SIO SHOULD BE A 24 BIT
12     DC.B $20,$20,$54,$48,$45,$20,$42,$49,$54,$20
13     DC.B $50,$41,$54,$54,$45,$52,$4E,$20,$53,$45
14     DC.B $4E,$54,$20,$46,$52,$4F,$4D,$20,$54,$48
15     DC.B $45,$20,$46,$44,$46,$20,$53,$49,$4F,$20
16     DC.B $53,$48,$4F,$55,$4C,$44,$20,$42,$45,$20
17     DC.B $41,$20,$32,$34,$20,$42,$49,$54
18     DC.B $0A,$0D      ; lf cr
19  * ALTERNATING I/O PATTERN FOR CHANNEL A AND A 56 BIT
20     DC.B $20,$20,$41,$4C,$54,$45,$52,$4E,$41,$54
21     DC.B $49,$4E,$47,$20,$31,$2F,$30,$20,$50,$41
22     DC.B $54,$54,$45,$52,$4E,$20,$46,$4F,$52,$20
23     DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$41,$20
24     DC.B $41,$4E,$44,$20,$41,$20,$35,$36,$20,$42
25     DC.B $49,$54
26     DC.B $0A,$0D      ; lf cr
27  * ALTERNATING I/O PATTERN FOR CHANNEL B THE DATA WILL BE
28     DC.B $20,$20,$41,$4C,$54,$45,$52,$4E,$41,$54
29     DC.B $43,$4E,$47,$20,$31,$2F,$30,$20,$50,$41
30     DC.B $54,$54,$45,$52,$4E,$20,$46,$4F,$52,$20
31     DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$42,$2E
32     DC.B $20,$20,$54,$48,$45,$20,$44,$41,$54,$41
33     DC.B $20,$57,$49,$4C,$4C,$20,$42,$45
34     DC.B $0A,$0D      ; lf cr
35  * COMPLEMENTED EVERY CYCLE THE TEK 318S1 CHANNELS ARE ASSIGNED
36     DC.B $20,$20,$43,$4F,$4D,$50,$4C,$45,$4D,$45
37     DC.B $4E,$54,$45,$44,$20,$45,$56,$45,$52,$59
38     DC.B $20,$43,$59,$43,$4C,$45,$52,$20,$20,$54
39     DC.B $48,$45,$20,$54,$45,$48,$20,$33,$31,$38
40     DC.B $53,$31,$20,$43,$48,$41,$4E,$4E,$45,$4C
41     DC.B $53,$20,$41,$52,$45,$20,$41,$53,$53,$49
42     DC.B $47,$4E,$45,$44
43     DC.B $0A,$0D      ; lf cr

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU5.DEF_NUM

```

44 * AS FOLLOWS:
45 DC.B $20,$20,$41,$53,$20,$46,$4F,$4C,$4C,$4F
46 DC.B $57,$53,$3A
47 DC.B $0A,$0D ; lf cr
48 * CHANNEL
49 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$43,$48,$41
50 DC.B $4E,$4E,$45,$4C,$20,$20,$20,$20,$20,$20,$20,$20
51 DC.B $53,$49,$47,$4E,$41,$4C
52 DC.B $0A,$0D ; lf cr
53 * CHANNEL A DATA
54 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$30,$20,$20,$20
55 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
56 DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$20,$41,$20
57 DC.B $44,$41,$54,$41
58 DC.B $0A,$0D ; lf cr
59 * CHANNEL A LOAD STROBE
60 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$31,$20,$20,$20
61 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
62 DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$20,$41,$20
63 DC.B $4C,$4F,$41,$44,$20,$53,$54,$52,$4F,$42
64 DC.B $45
65 DC.B $0A,$0D ; lf cr
66 * CHANNEL A BIT CLOCK (24 CYCLES, 250 KHZ)
67 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$32,$20,$20,$20
68 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
69 DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$20,$41,$20
70 DC.B $42,$49,$54,$20,$43,$4C,$4F,$43,$48,$20
71 DC.B $20,$28,$32,$34,$20,$43,$59,$43,$4C,$45
72 DC.B $53,$2C,$20,$32,$35,$30,$20,$4B,$48,$5A
73 DC.B $29
74 DC.B $0A,$0D ; lf cr
75 * CHANNEL B DATA
76 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$33,$20,$20,$20
77 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
78 DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$20,$42,$20
79 DC.B $44,$41,$54,$41
80 DC.B $0A,$0D ; lf cr
81 * CHANNEL B LOAD STROBE
82 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$34,$20,$20,$20
83 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
84 DC.B $43,$48,$41,$4E,$4E,$45,$4C,$20,$20,$42,$20
85 DC.B $4C,$4F,$41,$44,$20,$53,$54,$52,$4F,$42
86 DC.B $45
87 DC.B $0A,$0D ; lf cr
88 * CHANNEL B BIT CLOCK (56 CYCLES, 125 KHZ)
89 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$35,$20,$20,$20
90 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFSMENU5.DEF_NUM

```

91 DC.B $43,$48,$41,$4E,$45,$4C,$20,$42,$20
92 DC.B $42,$49,$54,$20,$43,$4C,$4F,$43,$48,$20
93 DC.B $20,$28,$35,$36,$20,$43,$58,$43,$4C,$45
94 DC.B $53,$2C,$31,$32,$35,$20,$48,$48,$5A
95 DC.B $29
96 DC.B $0A,$0D ; lf cr
97 * DATA IS VALID ON THE RISING EDGE OF THE BIT CLOCK WHEN THE LOAD
98 DC.B $20,$20,$44,$41,$54,$41,$20,$49,$53,$20
99 DC.B $56,$41,$4C,$49,$44,$20,$4F,$4E,$20,$54
100 DC.B $48,$45,$20,$52,$49,$53,$49,$4E,$47,$20
101 DC.B $45,$44,$47,$45,$20,$4F,$46,$20,$54,$48
102 DC.B $45,$20,$42,$48,$54,$20,$43,$4C,$4F,$43
103 DC.B $48,$20,$57,$48,$45,$4E,$20,$54,$48,$45
104 DC.B $20,$4C,$4F,$41,$44
105 DC.B $0A,$0D ; lf cr
106 * STROBE IS LOW. THE CLOCK RATE SHOULD BE 250 KHZ FOR CHANNEL A AND
107 DC.B $20,$20,$53,$54,$52,$4F,$42,$45,$20,$49
108 DC.B $53,$20,$4C,$4F,$57,$2E,$20,$20,$54,$48
109 DC.B $45,$20,$43,$4C,$4F,$43,$48,$20,$52,$41
110 DC.B $54,$45,$20,$53,$48,$4F,$55,$4C,$44,$20
111 DC.B $42,$45,$20,$32,$35,$30,$20,$4B,$48,$5A
112 DC.B $20,$46,$4F,$52,$20,$43,$48,$41,$4E,$4E
113 DC.B $45,$4C,$20,$41,$20,$41,$4E,$44
114 DC.B $0A,$0D ; lf cr
115 * 125 KHZ FOR CHANNEL B. THE LOAD STROBE SHOULD GO HIGH AT THE END
116 DC.B $20,$20,$31,$32,$35,$20,$4B,$48,$5A,$20
117 DC.B $46,$4F,$52,$20,$43,$48,$41,$4E,$4E,$45
118 DC.B $4C,$20,$42,$2E,$20,$20,$54,$48,$45,$20
119 DC.B $4C,$4F,$41,$44,$20,$53,$54,$52,$4F,$42
120 DC.B $45,$20,$53,$48,$4F,$55,$4C,$44,$20,$47
121 DC.B $4F,$20,$48,$49,$47,$48,$20,$41,$54,$20
122 DC.B $54,$48,$45,$20,$45,$4E,$44
123 DC.B $0A,$0D ; lf cr
124 * OF EACH CYCLE.
125 DC.B $20,$20,$4F,$46,$20,$45,$41,$43,$48,$20
126 DC.B $43,$59,$43,$4C,$45,$2E
127 DC.B $0A,$0D ; lf cr
128 DC.B $0A,$0D ; lf cr
129 * HIT THE APPROPRIATE KEY TO PROCEED
130 DC.B $20,$20,$48,$49,$54,$20,$54,$48,$45,$20
131 DC.B $41,$50,$50,$52,$4F,$50,$52,$49,$41,$54
132 DC.B $45,$20,$48,$45,$59,$20,$54,$4F,$20,$50
133 DC.B $52,$4F,$43,$45,$45,$44
134 DC.B $0A,$0D ; lf cr
135 * "P" - PASS
136 DC.B $20,$20,$20,$20,$22,$50,$22,$20,$20,$20
137 DC.B $20,$50,$41,$53,$53

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU5.DEF_NUM

```

138      DC.B S0A,S0D      ; lf cr
139      * "F" - FAIL (LOAD STROBE GOES LOW FOR BOTH CHANNELS AND DATA
140      DC.B S20,$20,$20,$20,$22,$46,$22,$20,$20,$20
141      DC.B S20,$46,$41,$49,$4C,$20,$20,$28,$4C,$4F
142      DC.B S41,$44,$20,$53,$54,$52,$4F,$42,$45,$20
143      DC.B S47,$4F,$45,$53,$20,$4C,$4F,$57,$20,$46
144      DC.B S4F,$52,$20,$42,$4F,$54,$48,$20,$43,$48
145      DC.B S41,$4E,$4E,$45,$4C,$53,$20,$41,$4E,$44
146      DC.B S20,$44,$41,$54,$41
147      DC.B S0A,S0D      ; lf cr
148      * "O" - OTHER
149      DC.B S20,$20,$20,$20,$20,$20,$20,$20,$20,$20
150      DC.B S20,$20,$20,$20,$20,$20,$20,$20,$20,$20
151      DC.B S43,$4B,$20,$50,$52,$45,$53,$45,$4E,$54
152      DC.B S20,$42,$55,$54,$20,$44,$41,$54,$41,$20
153      DC.B S4F,$46,$20,$45,$49,$54,$48,$45,$52,$2F
154      DC.B S42,$4F,$54,$48,$20,$43,$48,$41,$4E,$4E
155      DC.B S45,$4C,$53,$20,$49,$53,$20,$49,$4E,$43
156      DC.B S4F,$52,$52,$45,$43,$54,$29
157      DC.B S0A,S0D      ; lf cr
158      * "O" - OTHER
159      DC.B S20,$20,$20,$20,$20,$22,$4F,$22,$20,$20,$20
160      DC.B S20,$4F,$54,$48,$45,$52
161      *-> DFSMENU6 BEGIN
162      DC.B S00           ; end string

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFSMENU6.DEF_NUM

15.6 DFSMENU6.DEF_NUM

```

***** Source Listing --> DFSMENU6.DEF_NUM *****
*****
1  DS.W 0 ;word align menu
2  DFSMENU6 DS.W 0 ;label of menu
3  DC.B SOA,$0D ; lf cr
4  DC.B SOA,$0D ; lf cr
5  * FDF SIO / FDF CONTROLLER FAULT ISOLATION
6  DC.B $20,$20,$46,$44,$46,$20,$53,$49,$4F,$20
7  DC.B $2F,$20,$46,$44,$46,$20,$43,$4F,$4E,$54
8  DC.B $52,$4F,$4C,$4C,$45,$52,$20,$46,$41,$55
9  DC.B $4C,$54,$20,$49,$53,$4F,$4C,$41,$54,$49
10 DC.B $4F,$4E
11 DC.B SOA,$0D ; lf cr
12 DC.B SOA,$0D ; lf cr
13 * 1. POWER DOWN THE ADPU BY REMOVING P2 OF W3 FROM J15 OF THE ADPU,
14 DC.B $20,$20,$20,$20,$20,$20,$31,$2E,$20
15 DC.B $20,$50,$4F,$57,$45,$52,$20,$44,$4F,$57
16 DC.B $4E,$20,$54,$48,$45,$20,$41,$44,$50,$55
17 DC.B $20,$42,$59,$20,$52,$45,$4D,$4F,$56,$49
18 DC.B $4E,$47,$20,$50,$32,$20,$4F,$46,$20,$57
19 DC.B $33,$20,$46,$52,$4F,$4D,$20,$4A,$31,$35
20 DC.B $20,$4F,$46,$20,$54,$48,$45,$20,$41,$44
21 DC.B $50,$55,$2C
22 DC.B SOA,$0D ; lf cr
23 * 2. REMOVE THE ADPU BOTTOM COVER TO GAIN ACCESS TO THE BACKPLANE,
24 DC.B $20,$20,$20,$20,$20,$20,$32,$2E,$20
25 DC.B $20,$52,$45,$4D,$4F,$56,$45,$20,$54,$48
26 DC.B $45,$20,$41,$44,$50,$55,$20,$42,$4F,$54
27 DC.B $54,$4F,$4D,$20,$43,$4F,$56,$45,$52,$20
28 DC.B $54,$4F,$20,$47,$41,$49,$4E,$20,$41,$43
29 DC.B $43,$45,$53,$53,$20,$54,$4F,$20,$54,$48
30 DC.B $45,$20,$42,$41,$43,$48,$50,$50,$4C,$41,$4E
31 DC.B $45,$2C
32 DC.B SOA,$0D ; lf cr
33 * 3. REPLACE P2 OF W3 TO J15 OF THE ADPU.
34 DC.B $20,$20,$20,$20,$20,$20,$33,$2E,$20
35 DC.B $20,$52,$45,$50,$4C,$41,$43,$45,$20,$50
36 DC.B $32,$20,$4F,$46,$20,$57,$33,$20,$54,$4F
37 DC.B $20,$4A,$31,$35,$20,$4F,$46,$20,$54,$48
38 DC.B $45,$20,$41,$44,$50,$55,$2E
39 DC.B SOA,$0D ; lf cr
40 DC.B SOA,$0D ; lf cr
41 * HIT ANY KEY TO PROCEED
42 DC.B $20,$20,$48,$49,$54,$20,$41,$4F,$59,$20
43 DC.B $4B,$45,$59,$20,$54,$4F,$20,$50,$52,$4F

```

Improved GUARDRAIL V MC68000 'DF' Files
DRCL:[ALGO,IGR.MC68000_DF]DFSMENU6.DEF_NUM

```
44      DC.B   S43,S45,S45,S44  
45      => DFSMENU7 BEGIN  
46      DC.B   S00           ; end string
```

15.7 DFSMENU7.DEF_NUM

```

***** Source Listing --> DFSMENU7.DEF_NUM *****
*****
1 DS W 0 ;word align menu
2 DFSMENU7 DS W 0 ;label of menu
3 DC B S0A,$0D ; lf cr
4 DC B S0A,$0D ; lf cr
5 * INSTRUCTIONS FOR PROBING FDF SIO CARD
6 DC B $20,$20,$49,$4E,$53,$54,$52,$55,$43,$54
7 DC B $49,$4F,$4E,$53,$20,$46,$4F,$52,$20,$50
8 DC B $52,$4F,$42,$49,$4E,$47,$20,$46,$44,$46
9 DC B $20,$53,$49,$4F,$20,$43,$41,$52,$44
10 DC B S0A,$0D ; lf cr
11 DC B S0A,$0D ; lf cr
12 * THE FOLLOWING TABLE PRESENTS THE CONDITIONS THAT WOULD
13 DC B $20,$20,$54,$48,$45,$20,$46,$4F,$4C,$4C
14 DC B $4F,$57,$49,$4E,$47,$20,$54,$41,$42,$4C
15 DC B $45,$20,$50,$52,$45,$53,$45,$4E,$54,$53
16 DC B $20,$54,$48,$45,$20,$43,$4F,$4E,$44,$49
17 DC B $54,$49,$4F,$4E,$53,$20,$54,$48,$41,$54
18 DC B $20,$57,$4F,$55,$4C,$44
19 DC B S0A,$0D ; lf cr
20 * IMPLY THAT THE FDF CONTROLLER CARD IS FUNCTIONING
21 DC B $20,$20,$49,$4D,$50,$4C,$59,$20,$54,$48
22 DC B $41,$54,$20,$54,$48,$45,$20,$46,$44,$46
23 DC B $20,$43,$4F,$4E,$54,$52,$4F,$4C,$4C,$45
24 DC B $52,$20,$43,$41,$52,$44,$20,$49,$53,$20
25 DC B $46,$55,$4E,$43,$54,$49,$4F,$4E,$49,$4E
26 DC B $47
27 DC B S0A,$0D ; lf cr
28 * CORRECTLY GIVEN THAT ONE OR BOTH CHANNELS OF THE FDF SIO
29 DC B $20,$20,$43,$4F,$52,$52,$45,$43,$54,$4C
30 DC B $59,$20,$47,$49,$56,$45,$4E,$20,$54,$48
31 DC B $41,$54,$20,$4F,$4E,$45,$20,$4F,$52,$20
32 DC B $42,$4F,$54,$48,$20,$43,$48,$41,$4E,$4E
33 DC B $45,$4C,$53,$20,$4F,$46,$20,$54,$48,$45
34 DC B $20,$46,$44,$46,$20,$53,$49,$4F
35 DC B S0A,$0D ; lf cr
36 * HAVE FAILED.
37 DC B $20,$20,$48,$41,$56,$45,$20,$46,$41,$49
38 DC B $4C,$45,$44,$2E
39 DC B S0A,$0D ; lf cr
40 DC B S0A,$0D ; lf cr
41 * SIO FDF CONTROLLER
42 DC B $20,$20,$53,$49,$4F,$20,$20,$20,$20
43 DC B $20,$20,$20,$46,$44,$46,$20,$43,$4F,$4E

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFSMENU7.DEF_NUM

```

44 DC.B $54,$52,$4F,$4C,$45,$52,$20,$20,$20,$20
45 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
46 DC.B $20,$20,$46,$44,$46,$20,$43,$4F,$4E,$54
47 DC.B $52,$4F,$4C,$4C,$45,$52
48 DC.B $0A,$00 ; lf cr
49 * CHANNEL CH A LOAD STROBE CH B LOAD STROBE
50 DC.B $20,$20,$43,$48,$41,$4E,$4E,$45,$4C,$20
51 DC.B $20,$20,$20,$43,$48,$20,$41,$20,$4C,$4F
52 DC.B $41,$44,$20,$53,$54,$52,$4F,$42,$45,$20
53 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
54 DC.B $20,$20,$43,$48,$20,$42,$20,$4C,$4F,$41
55 DC.B $44,$20,$53,$54,$52,$4F,$42,$45
56 DC.B $0A,$00 ; lf cr
57 * FAILURE (SLOT A13 PIN 20) (SLOT A13 PIN 24)
58 DC.B $20,$20,$46,$41,$48,$4C,$55,$52,$45,$20
59 DC.B $20,$20,$20,$28,$53,$4C,$4F,$54,$20,$41
60 DC.B $31,$33,$20,$50,$49,$4E,$20,$32,$30,$29
61 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
62 DC.B $20,$20,$28,$53,$4C,$4F,$54,$20,$41,$31
63 DC.B $33,$20,$50,$49,$4E,$20,$32,$34,$29
64 DC.B $0A,$00 ; lf cr
65 DC.B $0A,$00 ; lf cr
66 * A HIGH LOW
67 DC.B $20,$20,$41,$20,$20,$20,$20,$20,$20,$20
68 DC.B $20,$20,$20,$48,$49,$47,$48,$20,$20,$20
69 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
70 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
71 DC.B $20,$20,$20,$20,$4C,$4F,$57
72 DC.B $0A,$00 ; lf cr
73 * B LOW HIGH
74 DC.B $20,$20,$42,$20,$20,$20,$20,$20,$20,$20
75 DC.B $20,$20,$20,$4C,$4F,$57,$20,$20,$20,$20
76 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
77 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
78 DC.B $20,$20,$20,$48,$49,$47,$48
79 DC.B $0A,$00 ; lf cr
80 * A AND B HIGH HIGH
81 DC.B $20,$20,$41,$20,$41,$4E,$44,$20,$42,$20
82 DC.B $20,$20,$20,$48,$49,$47,$48,$20,$20,$20
83 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
84 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$20,$20
85 DC.B $20,$20,$20,$48,$49,$47,$48
86 DC.B $0A,$00 ; lf cr
87 DC.B $0A,$00 ; lf cr
88 * IF THE TWO LOAD STROBE CONDITIONS ABOVE HAVE BEEN MET FOR
89 DC.B $20,$20,$49,$46,$20,$54,$48,$45,$20,$54
90 DC.B $57,$4F,$20,$4C,$4F,$41,$44,$20,$53,$54

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFSMENU7.DEF_NUM

```

91 DC.B $52,$4F,$42,$45,$20,$43,$4F,$4E,$44,$49
92 DC.B $54,$49,$4F,$4E,$53,$20,$41,$42,$4F,$56
93 DC.B $45,$20,$48,$41,$56,$45,$20,$42,$45,$45
94 DC.B $4E,$20,$4D,$45,$54,$20,$46,$4F,$52
95 DC.B $0A,$0D ; lf cr
96 * THE PARTICULAR FDF SIO FAILURE THEN THE FAULT HAS BEEN
97 DC.B $20,$20,$54,$48,$45,$20,$50,$41,$52,$54
98 DC.B $49,$43,$55,$4C,$41,$52,$20,$46,$44,$46
99 DC.B $20,$53,$49,$4F,$20,$46,$41,$49,$4C,$55
100 DC.B $52,$45,$20,$54,$48,$45,$4E,$20,$54,$48
101 DC.B $45,$20,$46,$41,$55,$4C,$54,$20,$48,$41
102 DC.B $53,$20,$42,$45,$45,$4E
103 DC.B $0A,$0D ; lf cr
104 * ISOLATED TO THE FDF SIO AND IT IS CONSIDERED TO HAVE FAILED.
105 DC.B $20,$20,$49,$53,$4F,$4C,$41,$54,$45,$44
106 DC.B $20,$54,$4F,$20,$54,$48,$45,$20,$46,$44
107 DC.B $46,$20,$53,$49,$4F,$20,$41,$4E,$44,$20
108 DC.B $49,$54,$20,$49,$53,$20,$43,$4F,$4E,$53
109 DC.B $49,$44,$45,$52,$45,$44,$20,$54,$4F,$20
110 DC.B $48,$41,$56,$45,$20,$46,$41,$49,$4C,$45
111 DC.B $44,$5E
112 DC.B $0A,$0D ; lf cr
113 * OTHERWISE THE FDF SIO TEST IS CONSIDERED TO HAVE PASSED.
114 DC.B $20,$20,$4F,$54,$48,$45,$52,$57,$49,$53
115 DC.B $45,$20,$54,$48,$45,$20,$46,$44,$46,$20
116 DC.B $53,$49,$4F,$20,$54,$45,$53,$54,$20,$49
117 DC.B $53,$20,$43,$4F,$4E,$53,$48,$44,$45,$52
118 DC.B $45,$44,$20,$54,$4F,$20,$48,$41,$56,$45
119 DC.B $20,$50,$41,$53,$53,$45,$44,$5E
120 DC.B $0A,$0D ; lf cr
121 DC.B $0A,$0D ; lf cr
122 * HIT THE APPROPRIATE KEY TO PROCEED
123 DC.B $20,$20,$48,$49,$54,$20,$54,$48,$45,$20
124 DC.B $41,$50,$50,$52,$4F,$50,$52,$49,$41,$54
125 DC.B $45,$20,$4B,$45,$59,$20,$54,$4F,$20,$50
126 DC.B $52,$4F,$43,$45,$45,$44
127 DC.B $0A,$0D ; lf cr
128 * "P" - PASS (FDF CONTROLLER NOT PROVIDING LOAD STROBE)
129 DC.B $20,$20,$20,$20,$20,$22,$50,$22,$20,$2D
130 DC.B $20,$50,$41,$53,$53,$20,$28,$46,$44,$46
131 DC.B $20,$43,$4F,$4E,$54,$52,$4F,$4C,$4C,$45
132 DC.B $52,$20,$4E,$4F,$54,$20,$50,$52,$4F,$56
133 DC.B $49,$44,$49,$4E,$47,$20,$4C,$4F,$41,$44
134 DC.B $20,$53,$54,$52,$4F,$42,$45,$29
135 DC.B $0A,$0D ; lf cr
136 * "F" - FAIL (FDF CONTROLLER IS PROVIDING LOAD STROBE)
137 DC.B $20,$20,$20,$20,$20,$22,$46,$22,$20,$2D

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU7.DEF_NUM

```

138 DC.B $20,$46,$41,$49,$4C,$20,$28,$46,$44,$46
139 DC.B $20,$43,$4F,$4E,$54,$52,$4F,$4C,$4C,$45
140 DC.B $52,$20,$49,$53,$20,$20,$50,$52,$4F,$56
141 DC.B $49,$44,$49,$4E,$47,$20,$4C,$4F,$41,$44
142 DC.B $20,$53,$54,$52,$4F,$42,$45,$29
143 DC.B $00 ; end string
144 DFSLINE6 DS.W 0 ;label of sub menu
145 DC.B $0A,$0D ; lf cr
146 DC.B $0A,$0D ; lf cr
147 * HIT "Q" TO QUIT
148 DC.B $20,$20,$48,$49,$54,$20,$22,$51,$22,$20
149 DC.B $54,$4F,$20,$51,$55,$49,$54
150 DC.B $0A,$0D ; lf cr
151 * OR "R" TO REPEAT
152 DC.B $20,$20,$20,$4F,$52,$20,$52,$52,$22,$20
153 DC.B $54,$4F,$20,$52,$45,$50,$45,$41,$54
154 DC.B $00 ; end string
155 DFSLINE7 DS.W 0 ;label of sub menu
156 DC.B $0A,$0D ; lf cr
157 DC.B $0A,$0D ; lf cr
158 * FDF SIO CARD (10-156022-3, A13) HAS FAILED.
159 DC.B $20,$20,$46,$44,$46,$20,$53,$49,$4F,$20
160 DC.B $43,$41,$52,$44,$20,$28,$31,$30,$20,$31
161 DC.B $35,$36,$30,$32,$32,$20,$33,$20,$20,$41
162 DC.B $31,$33,$29,$20,$48,$41,$53,$20,$46,$41
163 DC.B $49,$4C,$45,$44,$2E
164 DC.B $00 ; end string
165 DFSLINE8 DS.W 0 ;label of sub menu
166 DC.B $0A,$0D ; lf cr
167 DC.B $0A,$0D ; lf cr
168 * FDF CONTROLLER (10-158506-1, A25) HAS FAILED.
169 DC.B $20,$20,$46,$44,$46,$20,$43,$4F,$4E,$54
170 DC.B $52,$4F,$4C,$4C,$54,$52,$20,$28,$31,$30
171 DC.B $20,$31,$35,$38,$35,$30,$36,$20,$31,$2C
172 DC.B $20,$41,$32,$35,$29,$20,$48,$41,$53,$20
173 DC.B $46,$41,$49,$4C,$45,$44,$2E
174 DC.B $00 ; end string

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFSMENU8.DEF_NUM

15.8 DFSMENU8.DEF_NUM

```
***** Source Listing --> DFSMENU8.DEF_NUM *****
*****
1  DS.W 0 ;word align menu
2  DFSMENU8 DS.W 0 ;label of menu
3  DC.B $0A,$0D ; lf cr
4  DC.B $0A,$0D ; lf cr
5  * HIT "Q" TO QUIT
6  DC.B $20,$20,$20,$20,$20,$20,$48,$49,$54
7  DC.B $20,$22,$51,$22,$20,$54,$4F,$20,$51,$55
8  DC.B $49,$54
9  DC.B $0A,$0D ; lf cr
10 * OR "R" TO REPEAT
11 DC.B $20,$20,$20,$20,$20,$20,$20,$20,$4F,$52
12 DC.B $20,$22,$52,$22,$20,$54,$4F,$20,$52,$45
13 DC.B $50,$45,$41,$54
14 DC.B $0D ; end string
```

Improved GUARDRAIL V MC68000 'DF' Files

16 FILES WITH EXTENSION '.FMT'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFA1.FMT_NUM

16.1 DFA1.FMT_NUM

***** Source Listing --> DFA1.FMT_NUM *****

1 LLEN 116
 2 MODULE NAME

3
 4 DFA1.MOD

5
 6
 7 PROGRAMMER

8
 9 ETO

10
 11
 12 DESCRIPTION

13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43

The DF Request Input job receives DF Request messages from the Uplink Message Handler (AGIN). There are two states that a DF request will assume while in the ADPU. The "new" DF is the DF request that has just been received from the GDPU. The "current" DF is the DF request that is being processed. The DF request parameters will be made available as the new DF to be performed. DF segments will be synchronized by the receipt of new DF requests from the GDPU. This job starts the process on the next DF by scheduling the Start/Continue DF job (DFA2). The "done with current" flag indicates whether or not the last (current) DF completed prior to the receipt of the new request. During normal operation, the current DF should complete before the new DF request is received from the GDPU. If a DF request is received before the current DF has completed, an error message is generated to indicate the problem. The Nav. Data Input job (DFA4) is scheduled after a timeout to read the current navigation data during the middle of the DF data collection.

CALLING SEQUENCE

?SCHED DFA1,#2,...,IPACK

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO,IGR,MC68000_DF]DFA1.FMT_NUM

44 * IPACK Int**4/Input
 45 * Pointer to packet containing DF request.
 46 * Packet definition "DF".
 47 *

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

LOWERREFERENCES

48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *

16.2 DFA2.FMT_NUM

***** Source Listing --> DFA2.FMT_NUM *****

1 LLEN 116

2 * MODULE NAME

3 *

4 * DFA2.MOD

5 *

6 *

7 * PROGRAMMER

8 *

9 * ETO

10 *

11 *

12 *

13 * DESCRIPTION

14 *

15 *

16 *

17 * DFA2

18 *

This job sets up the commands for the DF hardware (ACU, RF processor, DF receiver) and initiates the Fast DF Controller to start the DF measurement. This job is invoked to start a DF or continue on an intermediate step of the current DF. When starting a DF, this job takes the DF parameters for the "new" DF and makes them the "current" DF. The hardware commands are set up, and the Fast DF Controller is initiated. When continuing the current DF, this job simply loads the hardware commands and initiates the Fast DF Controller. To start a new DF (the "next" DF) this job is scheduled by the DF Request Input job (DFA1). When continuing the next step of the current segment, this job is scheduled by the DF Data Input job (DFA3). When this job starts a new DF, it will clear the "done with current" flag to indicate that the DF has started.

CALLING SEQUENCE

?SCHED DFA2.#1,...

DISC FILES USED

43 *

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFA2.FMT_NUM

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

DFAL.MOD DF request input (ARF).

LOWER REFERENCES

DETBAND.MOD Determine frequency band

GETBITS.MOD Access BITS fields.

GETLPAK.MOD Allocate a large packet.

PUTPAK.MOD Release a small packet for reuse.

CLRACCS.MOD Clear the accumulators for DF data collection.

INITDF.MOD Initialize the DF data collection system.

STARTDF.MOD In'tiate (or buffer) a DF collection step.

WAITDF.MOD Wait for the current DF step to complete.

FMTRQ.MOD Format a frequency (5 bytes to 4 bytes).

SIOLOAD.MOD Load the SIO output buffer.

SIOSEND.MOD Prepare SIO for output with FDFC.

RFPSETUP.MOD Load a sequence of RF Processor commands.

ACUSETUP.MOD Set up the PIO with a sequence of ACU commands.

INQUEUE.MOD Enqueue items to identify the DF step.

DQUEUE.MOD Dequeue items that identify the DF step.

ADJFDFC.MOD Adjust FDFC command for bandwidth.

DMPFFIFO.MOD Empty FDFC fifo into a buffer.

ACCBUFF.MOD Accumulate DF data from a buffer.

Improved GUARDRAIL V MC68000 'DF' Files
DRG1:[ALGO,IGR,MC68000_DF]DFA2.FMT NUM

91 *	ACCUM.MOD	Accumulate DF baseline data
92 *		
93 *	SIGPRES.MOD	Determine if signal present from DF data.
94 *		
95 *	READAGC.MOD	Read AGC value from DF receiver.

16.3 DFA3.FMT_NUM

***** Source Listing --> DFA3.FMT_NUM *****

1 LLEN 116
 2 MODULE NAME

3
 4 DFA3.MOD

5
 6
 7 PROGRAMMER

8
 9 ETO

10
 11
 12 DESCRIPTION

13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43

This job is scheduled when the Fast DF Controller interrupts the ADPU indicating that the Fast DF Controller has completed the current step. This job removes data (baseline data) from the Fast DF Controller. After reading the data, based on parameters for the "current" DF, this job determines whether the data read in was for the last step of the current DF or if there are more steps for the current segment. If there are more steps, this job schedules the Start/Continue DF job (DFA2) after removing the data and adding it to the accumulations for the current segment. If the data was for the last step in a segment, this job will also schedule the appropriate job based on type of DF. If the request was a coarse DF, then Calculate LOB (DFA6) would be scheduled. If a fine or calibration request, then the accumulation job would be scheduled (DFA5). If the data read in was for the last step of a segment, this job sets the "done with current" flag. If the step that just completed was the last step of the first segment of a Fine or Calibration request, this job reads the AGC value. The AGC value is used for successive segments of the request.

CALLING SEQUENCE

?SCHED DFA3.#2...

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA3.FMT_NUM

44 *
 45 *
 46 *
 47 *
 48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *
 59 *
 60 *
 61 *
 62 *
 63 *
 64 *
 65 *
 66 *
 67 *
 68 *

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

LOWERREFERENCES

READACC.MOD Read ACC value from DF receiver.
 GETDATA.MOD Remove DF data from FDFC
 ACCUM.MOD Accumulate DF baseline data
 INFQUAD.MOD Calculate phase angle from in-phase and quadrature.
 DIRREV.MOD Average direct and reverse baseline measurements

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO: IGR.MC68000_DF]DFA4.FMT_NUM

16.4 DFA4.FMT_NUM

***** Source Listing --> DFA4.FMT_NUM *****

```

1  LLEN 116
2  MODULE NAME
3
4  DFA4 MOD
5
6
7  PROGRAMMER
8
9  ETO
10
11
12  DESCRIPTION
13
14
15
16  DFA4
17
18  The Navigation Data Timer is scheduled to take a reading of
19  the location (and other items) of the ARF during the DF
20  measurement process. DF request input job (DFA1) sets up a
21  wait so that the reading will be taken halfway
22  (approximately) through the segment's measurements. When the
23  wait is complete, the Nav. Data is read and the various
24  parameters of the current Nav. Data are updated.
25
26
27  CALLING SEQUENCE
28
29
30  ?SCHED DFA4,#3,...
31
32
33  DISC FILES USED
34
35
36  EXTERNAL AND COMMON REFERENCES
37
38
39  HIGHER REFERENCES
40
41
42  LOWERREFERENCES
43

```

Improved GUARDRAIL V MC68000 'DF' Files
DRCL:[ALGO.IGR.MC68000_DF]DFA4.FMT_NUM

44 *
45 *

READNAV.MOD Read navigation data.

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA5.FMT_NUM

16.5 DFA5.FMT_NUM

***** Source Listing ==> DFA5.FMT_NUM *****

1 LLEN 116
 2 MODULE NAME

3
 4 DFA5.MOD

5
 6 PROGRAMMER

7
 8 ETO

9
 10 DESCRIPTION

11
 12 DFA5

13 This job is scheduled when a Fine segment has been completed.
 14 The baseline data is added to the accumulations for the
 15 current Fine request. The current Nav. data is input and
 16 added to the Nav. data accumulations. If the current DF
 17 segment was the last segment of the current Fine request,
 18 Calculate LOB (DFA6) is scheduled. If a segment comes in out
 19 of order or the DF segment failed (bad segment), a message is
 20 generated to indicate an error.

21 CALLING SEQUENCE

22 ?SCHED DFA5,#3,,,IPACK

23 IPACK Int*4/Input

24 Pointer to packet containing baseline data.
 25 Packet definition "SD".

26 DISC FILES USED

27 EXTERNAL AND COMMON REFERENCES

Improved GUARDRAIL V MC68000 'DF' Files
DRG1:[ALGO.IGR.MC68000_DF]DFA5.FMT_NUM

44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *

HIGHER REFERENCES

LOWERREFERENCES

GETPAK.MOD	Allocate a small packet.
PUTPAK.MOD	Release a small packet for reuse.
INPQUAD.MOD	Calculate phase angle from in-phase and quadrature.
DIRREV.MOD	Average direct and reverse baseline measurements

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.FMT_NUM

16.6 DFA6.FMT_NUM

***** Source Listing ==> DFA6.FMT_NUM *****

1 LLEN 116

2 * MODULE NAME

3 *
 4 * DFA6.MOD
 5 *
 6 *
 7 *
 8 *
 9 *
 10 *
 11 *
 12 *
 13 *
 14 *
 15 *
 16 *
 17 *
 18 *
 19 *
 20 *
 21 *
 22 *
 23 *
 24 *
 25 *
 26 *
 27 *
 28 *
 29 *
 30 *
 31 *
 32 *
 33 *
 34 *
 35 *
 36 *
 37 *
 38 *
 39 *
 40 *
 41 *
 42 *
 43 *

DFA6.MOD

PROGRAMMER

ETO

DESCRIPTION

DFA6

This job is scheduled with either the baseline data for one segment, the accumulated data for a Fine DF, or the accumulated data for a Calibration DF (see DFA3,DFA5). This job takes the baseline data and calculates an LOB. If the input was from a coarse DF (one segment), the current Navigation data is read to give an LOP (for a calibration or fine DF, the Navigation data is taken during the middle segment of the data collection segments). If an LOP cannot be calculated, it is reflected in the DF response message.

CALLING SEQUENCE

?SCHED DFA6,#3,,,IPACK

IPACK

Int*4/Input

Pointer to packet containing baseline phase angles. Either packet definition "TFA" or "CDF".

DISC FILES USED

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFA6.FMT_NUM

EXTERNAL AND COMMON REFERENCES

44 *
 45 *
 46 *
 47 *
 48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *
 59 *
 60 *
 61 *
 62 *
 63 *
 64 *

HIGHER REFERENCES

LOWERREFERENCES

BCDBIN.MOD Convert BCD to binary integer.
 COARSE.MOD Coarse search of LOP calculation
 FMTDFR.MOD Format DF response message
 P2SRCH.MOD Pass 2 search of LOP calculation.
 PARFIT.MOD Three point parabolic fit.
 PUTPAK.MOD Release a small packet for reuse.
 GETPAK.MOD Allocate a small packet.

Improved GUARDRAIL V MC68000 'DF' Files
 DR01:[ALGO.IGR.MC68000_DF]DFALG.FMT_NUM

16.7 DFALG.FMT_NUM

```
***** Source Listing --> DFALG.FMT_NUM *****
*****
1 .EJECT
2 .BTITLE?ACCUM.MOD? BOTTOM ?PAGE ??
3 .TTITLE?ACCUM.MOD?DFALG.MST?PAGE ??
4 .PARA
5 .UNDERSORE
6 .INDENT -5
7 1.0 MODULE NAME
8 .PARA
9 .FILL
10 ACCUM.MOD
11 .PARA
12 .UNDERSORE
13 .INDENT -5
14 2.0 PROGRAMMER
15 .PARA
16 .FILL
17 ETO
18 .PARA
19 .UNDERSORE
20 .INDENT -5
21 3.0 DESCRIPTION
22 .PARA
23 .FILL
24 This module accumulates the new raw DF data with
25 the data collected so far for the current segment.
26 Since not all steps within a segment are identical,
27 this module must determine the order (which baselines,
28 direct and reverse) that the data is stored in the
29 buffer. Once the data is removed from the buffer, it
30 is added to the appropriate accumulator.
31 .PARA
32 .UNDERSORE
33 .INDENT -5
34 4.0 CALLING SEQUENCE
35 .PARA
36 .FILL
37 .skip
38 .in 10
39 JSR ACCUM
40 .skip
41 .PARA
42 .LMARGIN +17
43 .BLOCK BEGIN
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFALG.FMT_NUM

```

44 .INDENT -19
45 Dn
46 .BREAK
47 Segment number of current DF
48 .PARA
49 .BLOCK END
50 .BLOCK BEGIN
51 .INDENT -19
52 Dn
53 .BREAK
54 Step number of current step of current segment.
55 .PARA
56 .BLOCK END
57 .BLOCK BEGIN
58 .INDENT -19
59 Dn
60 .BREAK
61 Frequency band code (1-3)
62 .PARA
63 .BLOCK END
64 .BLOCK BEGIN
65 .INDENT -19
66 An
67 .BREAK
68 Pointer to buffer containing the data that was just
69 collected.
70 .PARA
71 .BLOCK END
72 .BLOCK BEGIN
73 .INDENT -19
74 An
75 .BREAK
76 Pointer to array of accumulators.
77 .BLOCK END
78 .LMARGIN RESET
79 .PARA
80 .UNDERScore
81 .INDENT -5
82 5.0 DISC FILES USED
83 .PARA
84 .UNDERScore
85 .INDENT -5
86 6.0 EXTERNAL AND COMMON REFERENCES
87 .PARA
88 .UNDERScore
89 .INDENT -5
90 7.0 HIGHER REFERENCES

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO.IGR.MC68000_DF]DFALG.FMT_NUM

```

91 .FILL
92 .PARA
93 .LMARGIN +15
94 .INDENT -15
95 DFA3.MOD
96 DF data input.
97 .LMARGIN RESET
98 .PARA
99 .UNDERSCORE
100 .INDENT -5
101 8.0 LOWER REFERENCES
102 .EJECT
103 .TITLE?ADFINIT.MOD? BOTTOM ?PAGE ??
104 .TITLE?ADFINIT.MOD?DFALG.MST?PAGE ??
105 .PARA
106 .UNDERSCORE
107 .INDENT -5
108 1.0 MODULE NAME
109 .PARA
110 .FILL
111 ADFINIT.MOD
112 .PARA
113 .UNDERSCORE
114 .INDENT -5
115 2.0 PROGRAMMER
116 .PARA
117 .FILL
118 Semplinski
119 .PARA
120 .UNDERSCORE
121 .INDENT -5
122 3.0 DESCRIPTION
123 .PARA
124 .FILL
125 This routine is responsible for downloading PROM and
126 scheduling certain jobs at initialization.
127 .PARA
128 .UNDERSCORE
129 .INDENT -5
130 4.0 CALLING SEQUENCE
131 .PARA
132 .FILL
133 .in+10
134 ?sched ADFINIT,#2,...
135 .FILL
136 .PARA
137 .UNDERSCORE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFALG.FMT_NUM

```

138 .INDENT -5
139 5.0 DISC FILES USED
140 .PARA
141 .UNDERScore
142 .INDENT -5
143 6.0 EXTERNAL AND COTTON REFERENCES
144 .PARA
145 .UNDERScore
146 .INDENT -5
147 7.0 HIGHER REFERENCES
148 .PARA
149 .UNDERScore
150 .INDENT -5
151 8.0 LOWER REFERENCES
152 .EJECT
153 .TITLE?ARCTAN.MOD? BOTTOM ?PAGE Z?
154 .TITLE?ARCTAN.MOD?DFALG.MST?PAGE Z?
155 .PARA
156 .UNDERScore
157 .INDENT -5
158 1.0 MODULE NAME
159 .PARA
160 .FILL
161 ARCTAN.MOD
162 .PARA
163 .UNDERScore
164 .INDENT -5
165 2.0 PROGRAMMER
166 .PARA
167 .FILL
168 ETO
169 .PARA
170 .UNDERScore
171 .INDENT -5
172 3.0 DESCRIPTION
173 .PARA
174 .FILL
175 This module performs the arctangent function
176 of the given value.
177 .PARA
178 .UNDERScore
179 .INDENT -5
180 4.0 CALLING SEQUENCE
181 .PARA
182 .FILL
183 .skip
184 .in 10

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO,IGR,MC68000_DF]DFALG.FMT_NUM

```

185 JSR   ARCTAN
186 .SKIP
187 .PARA
188 .LMARGIN +17
189 .BLOCK BEGIN
190 .INDENT -19
191 Dn
192 .BREAK
193 .Argument for arctan.
194 .PARA
195 .BLOCK END
196 .BLOCK BEGIN
197 .INDENT -19
198 Dn
199 .BREAK
200 .Arctan of value
201 .BLOCK END
202 .LMARGIN RESET
203 .PARA
204 .UNDERSCORE
205 .INDENT -5
206 5.0 DISC FILES USED
207 .PARA
208 .UNDERSCORE
209 .INDENT -5
210 6.0 EXTERNAL AND COMMON REFERENCES
211 .PARA
212 .UNDERSCORE
213 .INDENT -5
214 7.0 HIGHER REFERENCES
215 .FILL
216 .PARA
217 .LMARGIN +15
218 .INDENT -15
219 INPQUAD.MOD
220 Calculate phase angle from in-phase and quadrature.
221 .LMARGIN RESET
222 .PARA
223 .UNDERSCORE
224 .INDENT -5
225 8.0 LOWER REFERENCES
226 .EJECT
227 .TITLE?COARSE.MOD? BOTTOM ?PAGE 1?
228 .TITLE?COARSE.MOD?DFALG.MST?PAGE 1?
229 .PARA
230 .UNDERSCORE
231 .INDENT -5

```

Int*4/Input

Int*4/Output

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFALG.FMT NUM

```

232 1.0  MODULE NAME
233 . PARA
234 . FILL
235 COARSE MOD
236 . PARA
237 . UNDERSORE
238 . INDENT -5
239 2.0  PROGRAMMER
240 . PARA
241 . FILL
242 ETO
243 . PARA
244 . UNDERSORE
245 . INDENT -5
246 3.0  DESCRIPTION
247 . PARA
248 . FILL
249 This module performs the coarse (or first pass)
250 search of the Cal tables for the LOP calculation.
251 The LOP calculation algorithm is made up of
252 two searches of the Cal tables for a match of the
253 measured phase angle and the angle of arrival.
254 The first pass compares the measured phase angle
255 with the value at each of the 180 nodes of the
256 Cal table for each baseline. The Cal tables are
257 interpolated over frequency to perform the
258 calculation. The purpose of the first pass search
259 is to determine a window of ADA's where the closest
260 match of the measured phase angles and the Cal table
261 values lies.
262 . PARA
263 . UNDERSORE
264 . INDENT -5
265 4.0  CALLING SEQUENCE
266 . PARA
267 . FILL
268 . skip
269 . in 10
270 JSR COARSE
271 . SKIP
272 . PARA
273 .LMARGIN +17
274 .BLOCK BEGIN
275 .INDENT -19
276 Dn
277 .BREAK
278 DF frequency in integer.
  
```

Int*4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFALG.FMT NUM

```

279 . PARA
280 . BLOCK END
281 . BLOCK BEGIN
282 . INDENT -19
283 Dn
284 . BREAK
285 Frequency band code (1-3)
286 . PARA
287 . BLOCK END
288 . BLOCK BEGIN
289 . INDENT -19
290 An
291 . BREAK
292 Pointer to buffer containing phase angle measurements
293 for each baseline.
294 . PARA
295 . BLOCK END
296 . BLOCK BEGIN
297 . INDENT -19
298 Dn
299 . BREAK
300 Index of AOA about which the window was determined.
301 . PARA
302 . BLOCK END
303 . BLOCK BEGIN
304 . INDENT -19
305 Dn
306 . BREAK
307 Status of first pass search.
308 . BLOCK END
309 .LMARGIN RESET
310 . PARA
311 . UNDERSCORE
312 . INDENT -5
313 5.0 DISC FILES USED
314 . PARA
315 . UNDERSCORE
316 . INDENT -5
317 6.0 EXTERNAL AND COMMON REFERENCES
318 . PARA
319 . UNDERSCORE
320 . INDENT -5
321 7.0 HIGHER REFERENCES
322 . FILL
323 . PARA
324 .LMARGIN +15
325 . INDENT -15
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFALG.FMT_NUM

```

326 DFA6 MOO
327 LOP calculation
328 LMARGIN RESET
329 . PARA
330 UNDERSOORE
331 .INDENT -5
332 8.0 LOWER REFERENCES
333 .EJECT
334 .TITLE?DET BAND MOO? BOTTOM PAGE 1?
335 .TITLE?DET BAND MOO?DFALG.MST?PARA 1?
336 . PARA
337 UNDERSOORE
338 .INDENT -5
339 1.0 MODULE NAME
340 . PARA
341 .FILL
342 DETBAND.MOO
343 . PARA
344 UNDERSOORE
345 .INDENT -5
346 2.0 PROGRAMMER
347 . PARA
348 .FILL
349 ETO
350 . PARA
351 UNDERSOORE
352 .INDENT -5
353 3.0 DESCRIPTION
354 . PARA
355 .FILL
356 This module determines which of the three frequency
357 bands the given frequency is within. The frequency
358 is converted from BCD to integer (binary), and
359 then tested against the boundaries of each frequency
360 band until a frequency band is determined.
361 . PARA
362 UNDERSOORE
363 .INDENT -5
364 4.0 CALLING SEQUENCE
365 . PARA
366 .FILL
367 .skip
368 .in 10
369 JSR DETBAND
370 .skip
371 . PARA
372 LMARGIN +17

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFALG.FMT_NUM

```

373 .BLOCK BEGIN
374 .INDENT -19
375 .An
376 .BREAK
377 Pointer to frequency (frequency in BCD).
378 .PARA
379 .BLOCK END
380 .BLOCK BEGIN
381 .INDENT -19
382 .Dn
383 .BREAK
384 Frequency band code (1-3).
385 .BLOCK END
386 .LMARGIN RESET
387 .PARA
388 .UNDERScore
389 .INDENT -5
390 5.0 DISC FILES USED
391 .PARA
392 .UNDERScore
393 .INDENT -5
394 6.0 EXTERNAL AND COMMON REFERENCES
395 .PARA
396 .UNDERScore
397 .INDENT -5
398 7.0 HIGHER REFERENCES
399 .FILL
400 .PARA
401 .LMARGIN +15
402 .INDENT -15
403 DFA2.MOD
404 Start/continue DF data collection
405 .LMARGIN RESET
406 .PARA
407 .UNDERScore
408 .INDENT -5
409 8.0 LOWER REFERENCES
410 .FILL
411 .PARA
412 .LMARGIN +15
413 .INDENT -15
414 BCDBIN.MOD
415 Convert BCD to binary integer
416 .LMARGIN RESET
417 .EJECT
418 .BTITLE?DFA1.MOD? BOTTOM ?PAGE ??
419 .TTITLE?DFA1.MOD?DFALG.MST?PAGE ??
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [A]GO.IGR.MC68000_DF]DFALG.FMT_NUM

```

420 . PARA
421 . UNDERSORE
422 . INDENT -5
423 1.0 MODULE NAME
424 . PARA
425 . FILL
426 DFAL.MOD
427 . PARA
428 . UNDERSORE
429 . INDENT -5
430 2.0 PROGRAMMER
431 . PARA
432 . FILL
433 ETO
434 . PARA
435 . UNDERSORE
436 . INDENT -5
437 3.0 DESCRIPTION
438 . PARA
439 . FILL
440 . imbed DFAL.STR//:
441 . PARA
442 . UNDERSORE
443 . INDENT -5
444 4.0 CALLING SEQUENCE
445 . PARA
446 . FILL
447 . skip
448 . in 10
449 ?SCHED DFAL.#2,...IPACK
450 . SKIP
451 . PARA
452 .LMARGIN +17
453 .BLOCK BEGIN
454 . INDENT -19
455 .IPACK
456 .BREAK
457 Pointer to packet containing DF request. Packet
458 definition "DF".
459 .BLOCK END
460 .LMARGIN RESET
461 . PARA
462 . UNDERSORE
463 . INDENT -5
464 5.0 DISC FILES USED
465 . PARA
466 . UNDERSORE

```

Int*4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2G.FMT_NUM

```

467 .INDENT -5
468 6.0 EXTERNAL AND COMMON REFERENCES
469 .PARA
470 .UNDERScore
471 .INDENT -5
472 7.0 HIGHER REFERENCES
473 .PARA
474 .UNDERScore
475 .INDENT -5
476 8.0 LOWER REFERENCES
477 .EJECT
478 .BTITLE?DFA2.MOD' BOTTOM ?PAGE 1?
479 .TTITLE?DFA2.MOD'DFALG.MST?PAGE 1?
480 .PARA
481 .UNDERScore
482 .INDENT -5
483 1.0 MODULE NAME
484 .PARA
485 .FILL
486 DFA2.MOD
487 .PARA
488 .UNDERScore
489 .INDENT -5
490 2.0 PROGRAMMER
491 .PARA
492 .FILL
493 ETO
494 .PARA
495 .UNDERScore
496 .INDENT -5
497 3.0 DESCRIPTION
498 .PARA
499 .FILL
500 .imbed DFA2.STR/G
501 .PARA
502 .UNDERScore
503 .INDENT -5
504 4.0 CALLING SEQUENCE
505 .PARA
506 .FILL
507 .skip
508 .in 10
509 ?SCHED DFA2.#1,...
510 .SKIP
511 .FILL
512 .PARA
513 .UNDERScore

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFALG.FMT_NUM

```

514 .INDENT -5
515 5.0 DISC FILE USED
516 .PARA
517 .UNDERSCORE
518 .INDENT -5
519 6.0 EXTERNAL AND COMMON REFERENCES
520 .PARA
521 .UNDERSCORE
522 .INDENT -5
523 7.0 HIGHER REFERENCES
524 .PARA
525 .UNDERSCORE
526 .INDENT -5
527 8.0 LOWER REFERENCES
528 .FILL
529 .PARA
530 .LMARGIN +15
531 .INDENT -15
532 DETBAND.MOD
533 Determine frequency band
534 .LMARGIN RESET
535 .PARA
536 .LMARGIN +15
537 .INDENT -15
538 SETUP.MOD
539 Set up commands for DF data collection.
540 .LMARGIN RESET
541 .PARA
542 .LMARGIN +15
543 .INDENT -15
544 STRTDF.MOD
545 Start DF measurement sequence.
546 .LMARGIN RESET
547 .EJECT
548 .BTITLE?DFA3.MOD? BOTTOM ?PAGE ??
549 .TTITLE?DFA3.MOD?DFALG.MST?PAGE ??
550 .PARA
551 .UNDERSCORE
552 .INDENT -5
553 1.0 MODULE NAME
554 .PARA
555 .FILL
556 DFA3.MOD
557 .PARA
558 .UNDERSCORE
559 .INDENT -5
560 2.0 PROGRAMMER

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO. IGR. MC68000_DF]DFALG.FMT NUM

```

561 . PARA
562 . FILL
563 ETO
564 . PARA
565 . UNDERSORE
566 . INDENT -5
567 3.0 DESCRIPTION
568 . PARA
569 . FILL
570 . Imbed DFA3 STR/C
571 . PARA
572 . UNDERSORE
573 . INDENT -5
574 4.0 CALLING SEQUENCE
575 . PARA
576 . FILL
577 . skip
578 . In 10
579 ?SCHED DFA3, #2, ...
580 . SKIP
581 . FILL
582 . PARA
583 . UNDERSORE
584 . INDENT -5
585 5.0 DISC FILES USED
586 . PARA
587 . UNDERSORE
588 . INDENT -5
589 6.0 EXTERNAL AND COMMON REFERENCES
590 . PARA
591 . UNDERSORE
592 . INDENT -5
593 7.0 HIGHER REFERENCES
594 . PARA
595 . UNDERSORE
596 . INDENT -5
597 8.0 LOWER REFERENCES
598 . FILL
599 . PARA
600 .LMARGIN +15
601 .INDENT -15
602 READACC.MOD
603 Read AGC value from DF receiver.
604 .LMARGIN RESET
605 . PARA
606 .LMARGIN +15
607 .INDENT -15

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: |ALGO. IGR. MC68000 DF|DFALG.FMT NUM

```

608 GETDATA MOD
609 Remove DF data from FNFC
610 LMARGIN RESET
611 PARA
612 LMARGIN +15
613 INDENT -15
614 ACCUM MOD
615 Accumulate DF baseline data
616 LMARGIN RESET
617 PARA
618 LMARGIN +15
619 INDENT -15
620 INQUAD MOD
621 Calculate phase angle from in-phase and quadrature.
622 LMARGIN RESET
623 PARA
624 LMARGIN +15
625 INDENT -15
626 DIRREV MOD
627 Average direct and reverse baseline measurements
628 LMARGIN RESET
629 EJECT
630 BTITLE?DFA4 MOD? BOTTOM ?PAGE ??
631 TTITLE?DFA4 MOD?FALG.NST PAGE ??
632 PARA
633 UNDERSCORE
634 INDENT -5
635 1.0 MODULE NAME
636 PARA
637 FILL
638 DFA4 MOD
639 PARA
640 UNDERSCORE
641 INDENT -5
642 2.0 PROGRAMMER
643 PARA
644 FILL
645 ETO
646 PARA
647 UNDERSCORE
648 INDENT -5
649 3.0 DESCRIPTION
650 PARA
651 FILL
652 Imbed DFA4 STR/C
653 PARA
654 UNDERSCORE

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFALG.FMT NUM

```

655 .INDENT -5
656 4 0 CALLING SEQUENCE
657 .PARA
658 .FILL
659 .SKIP
660 .in 10
661 ?SCHED DFA4.#3...
662 .SKIP
663 .FILL
664 .PARA
665 .UNDERScore
666 .INDENT -5
667 5 0 DISC FILE USED
668 .PARA
669 .UNDERScore
670 .INDENT -5
671 6 0 EXTERNAL AND COMMON REFERENCES
672 .PARA
673 .UNDERScore
674 .INDENT -5
675 7 0 HIGHER REFERENCES
676 .PARA
677 .UNDERScore
678 .INDENT -5
679 8 0 LOWER REFERENCES
680 .FILL
681 .PARA
682 .LMARGIN +15
683 .INDENT -15
684 READNAV.MOD
685 Read navigation data.
686 .LMARGIN RESET
687 .EJECT
688 .BTITLE?DFAS.MOD BOTTOM PAGE ??
689 .TTITLE?DFAS.MOD DFALG.MST'PAGE 17
690 .PARA
691 .UNDERScore
692 .INDENT -5
693 1 0 MODULE NAME
694 .PARA
695 .FILL
696 DFA5.MOD
697 .PARA
698 .UNDERScore
699 .INDENT -5
700 2 0 PROGRAMME
701 .PARA

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR.MC68000_DF]DFALG.FMT NUM

```

702 FILL
703 ETO
704 . PARA
705 . UNDERSCORE
706 . INDENT -5
707 3 0 . DESCRPT: V
708 . PARA
709 . FILL
710 . imbed DFA5 STR
711 . PARA
712 . UNDERSCORE
713 . INDENT -5
714 4 0 . CALLING F JUENCE
715 . PARA
716 . FILL
717 . skip
718 . in 10
719 ?SCHED DFA5,03 . IPACK
720 . SKIP
721 . PARA
722 .LMARGIN +17
723 .BLOCK BEGIN
724 . INDENT -19
725 . IPACK
726 . BREAK
727 Pointer to packe' containing baseline data Packet
728 definition "SD"
729 .BLOCK END
730 .LMARGIN RESET
731 . PARA
732 . UNDERSCORE
733 . INDENT -5
734 5 0 . DISC FILE: USED
735 . PARA
736 . UNDERSCORE
737 . INDENT -5
738 6 0 . EXTERNAL AND COMMON REFERENCES
739 . PARA
740 . UNDERSCORE
741 . INDENT -5
742 7 0 . HIGHER REFERENCES
743 . PARA
744 . UNDERSCORE
745 . INDENT -5
746 8 0 . LOWER REFERENCES
747 . FILL
748 . PARA
  
```

Int*4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFALG.FMT NUM

```

749 .LMARGIN +15
750 .INDENT -15
751 GETPAK MOD
752 Allocate a small packet.
753 .LMARGIN RESET
754 .PARA
755 .LMARGIN +15
756 .INDENT -15
757 PUTPAK MOD
758 Release a small packet for reuse.
759 .LMARGIN RESET
760 .PARA
761 .LMARGIN +15
762 .INDENT -15
763 INFOUAD MOD
764 Calculate phase angle from in-phase and quadrature.
765 .LMARGIN RESET
766 .PARA
767 .LMARGIN +15
768 .INDENT -15
769 DIRREV MOD
770 Average direct and reverse baseline measurements
771 .LMARGIN RESET
772 .EJECT
773 .BTITLE?DFA6.MOD: BOTTOM ?PAGE ??
774 .TTITLE?DFA6.MOD:DFALG.MST PAGE ??
775 .PARA
776 .UNDERScore
777 .INDENT -5
778 1.0 MODULE NAME
779 .PARA
780 FILL
781 DFA6 MOD
782 .PARA
783 .UNDERScore
784 .INDENT -5
785 2.0 PROGRAMMER
786 .PARA
787 FILL
788 ETO
789 .PARA
790 .UNDERScore
791 .INDENT -5
792 3.0 DESCRIPTION
793 .PARA
794 FILL
795 Imbed DFA6 STR//

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFALG.FMT NUM

```

796 . PARA
797 . UNDERSORE
798 . INDENT -5
799 4 0 CALLING SEQUENCE
800 . PARA
801 . FILL
802 . SKIP
803 . IN 10
804 ?SCHED DFA6,03...IPACK
805 . SKIP
806 . PARA
807 .LMARGIN +17
808 . BLOCK BEGIN
809 . INDENT -19
810 . IPACK
811 . BREAK
812 Pointer to packet containing baseline phase
813 angles. Either packet definition "TFA" or
814 "CDF"
815 . BLOCK END
816 .LMARGIN RESET
817 . PARA
818 . UNDERSORE
819 . INDENT -5
820 5.0 DISC FILES USED
821 . PARA
822 . UNDERSORE
823 . INDENT -5
824 6.0 EXTERNAL AND COMMON REFERENCES
825 . PARA
826 . UNDERSORE
827 . INDENT -5
828 7.0 HIGHER REFERENCES
829 . PARA
830 . UNDERSORE
831 . INDENT -5
832 8.0 LOWER REFERENCES
833 . FILL
834 . PARA
835 .LMARGIN +15
836 . INDENT -15
837 BCDBIN MOD
838 Convert BCD to binary integer
839 .LMARGIN RESET
840 . PARA
841 .LMARGIN +15
842 . INDENT -15
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFALG.FMT NUM

```

843 COARSE MOD
844 Coarse search of TOP calculation
845 LMARGIN RESET
846 . PARA
847 LMARGIN +15
848 .INDENT -15
849 FMTDFR MOD
850 Format DF response message
851 LMARGIN RESET
852 . PARA
853 LMARGIN +15
854 .INDENT -15
855 P2SRCH MOD
856 Pass 2 search of TOP calculation
857 LMARGIN RESET
858 . PARA
859 LMARGIN +15
860 .INDENT -15
861 PARFIT MOD
862 Three point parabolic fit
863 LMARGIN RESET
864 . PARA
865 LMARGIN +15
866 .INDENT -15
867 PUTPAK MOD
868 Release a small packet for reuse
869 LMARGIN RESET
870 . PARA
871 LMARGIN +15
872 .INDENT -15
873 GETPAK MOD
874 Allocate a small packet
875 LMARGIN RESET
876 .EJECT
877 .BTITLE?DIRREV.M?D? BOTTOM ?PAGE ??
878 .TTITLE?DIRREV.M?D?DFALG.MST?PAGE ??
879 . PARA
880 .UNDERSCORE
881 .INDENT -5
882 1.0 MODULE NAME
883 . PARA
884 .FILL
885 .DIRREV MOD
886 . PARA
887 .UNDERSCORE
888 .INDENT -5
889 2.0 PROGRAMMER

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1: [ALGO. IGR.MC68000_DF]DFALG.FMT NUM

```

890 PARA
891 FILL
892 ETO
893 PARA
894 UNDERSCORE
895 INDENT -5
896 3 0 DESCRIPTOR
897 PARA
898 FILL
899 This module averages the direct and reverse baseline
900 phase angle measurements. If measurements are collected
901 for direct and reverse direction for each baseline.
902 Before LOP calculation is performed, the direct and reverse
903 measurements are averaged to give a single phase angle for
904 the baseline.
905 The formula to average is
906  $\text{no fill}$ 
907  $\text{Phase angle} = (\text{angle dir.} + (-\text{angle rev.})) / 2$ 
908  $\text{fill}$ 
909 PARA
910 UNDERSCORE
911 INDENT -5
912 4 0 CALLING SEQUENCE
913 PARA
914 FILL
915 skip
916 In 10
917 JSR DIRREV
918 skip
919 PARA
920 LMARGIN +17
921 BLOCK BEGIN
922 INDENT -19
923 Dn
924 BREAK
925 Frequency band code (1-3) Used to determine
926 total number of baselines
927 PARA
928 BLOCK END
929 BLOCK BEGIN
930 INDENT -19
931 An
932 BPAK
933 Pointer to buffer containing direct and reverse
934 phase angles for each baseline. Buffer is
935 organized such that the direct angle is followed
936 by the reverse angle for the baseline

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.1GR.MC68000_DF]DFALG.FMT NUM

```

917 . PARA
918 . BLOCK END
919 . BLOCK BEGIN
920 . INDENT -13
921 . An
922 . BREAK
923 . Pointer to buffer where the single phase analysis
924 . for each baseline are to be stored
925 . BLOCK END
926 . LMARGIN RESET
927 . PARA
928 . UNDERScore
929 . INDENT -5
930 . 5 0 DISC FILE USED
931 . PARA
932 . UNDERScore
933 . INDENT -5
934 . 6 0 EXTERNAL AND COMMON REFERENCES
935 . PARA
936 . UNDERScore
937 . INDENT -5
938 . 7 0 HIGHER REFERENCES
939 . FILL
940 . PARA
941 . LMARGIN +15
942 . INDENT -15
943 . DFA1 MOD
944 . DF data input.
945 . LMARGIN RESET
946 . PARA
947 . LMARGIN +15
948 . INDENT -15
949 . DFA5 MOD
950 . Accumulate data for Fine Ref.
951 . LMARGIN RESET
952 . PARA
953 . UNDERScore
954 . INDENT -5
955 . 8 0 LOWER REFERENCES
956 . EFFECT
957 . RTITLE:PMIDFR M-02 BOTTOM 2PA-12
958 . TITLE:PMIDFR M-02DFALG M-12PA-12
959 . PARA
960 . UNDERScore
961 . INDENT -5
962 . 1 0 MODULE NAME
963 . PARA

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR.MC68000_DF]DFALG.FMT NUM

```

984 FILE
985 FMTDEF MOD
986 PARA
987 UNDERSCORE
988 INDENT 5
989 2 0 PROGRAM
990 PARA
991 FILE
992 ETO
993 PARA
994 UNDERSCORE
995 INDENT 5
996 3 0 DESCRIPTI
997 PARA
998 FILE
999 This module take the data from the input packet and
1000 formats a DF response message to the output packet.
1001 PARA
1002 UNDERSCORE
1003 INDENT 5
1004 4 0 CALLING SEQUENCE
1005 PARA
1006 FILE
1007 skip
1008 in 10
1009 JSR FMTDEF
1010 skip
1011 PARA
1012 LMARGIN +17
1013 BLOCK BEGIN
1014 INDENT -19
1015 An
1016 BREAK
1017 Pointer to packet containing DF baseline data
1018 Either packet definition TFA or CDF
1019 PARA
1020 BLOCK END
1021 BLOCK BEGIN
1022 INDENT -19
1023 An
1024 BREAK
1025 Pointer to packet to contain the DF response
1026 message
1027 BLOCK END
1028 LMARGIN RESET
1029 PARA
1030 UNDERSCORE

```

Input/Output

Input/Output

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO:IGR:MC68000_DF]DFALG.FMT NUM

```

1031 INENT -5
1032 S 3 DISC FILE USED
1033 PARA
1034 UNDERSCORE
1035 INENT -5
1036 6 0 EXTERNAL NO COMMENT REFERENCES
1037 PARA
1038 UNDERSCORE
1039 INENT -5
1040 7 0 HIGHER REFERENCES
1041 FILE
1042 PARA
1043 LMARGIN +15
1044 INENT -15
1045 DFAP MOD
1046 LOP calculation
1047 LMARGIN RESET
1048 PARA
1049 UNDERSCORE
1050 INENT -5
1051 8 0 LOWER REFERENCES
1052 EFFECT
1053 RTT:GETDATA MOD? ROTTER 2PAGE 12
1054 TTT:GETDATA MOD?DFALG 1STPAGE 12
1055 PARA
1056 UNDERSCORE
1057 INENT -5
1058 1 0 MODULE NAME
1059 PARA
1060 FILE
1061 GETDATA MOD
1062 PARA
1063 UNDERSCORE
1064 INENT -5
1065 2 0 PROGRAM
1066 PARA
1067 FILE
1068 ETO
1069 PARA
1070 UNDERSCORE
1071 INENT -5
1072 3 0 DESCRIPTION
1073 PARA
1074 FILE
1075 This module removes the current raw baseline data
1076 from the Fast DF controller. See each step of
1077 measurement sequence is not identical, a different number
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO: IGR: MC68000_DF]DFALG: FMT NUM

1078 of DF measurement must be removed after each step. This
 1079 module determine how many measurements to push, and then
 1080 reals then into a buffer.

1081 PARA
 1082 UNDERSCORE
 1083 INDENT -5
 1084 4 0 CALLING SILENCE

1085 PARA
 1086 FILL
 1087 skip

1088 in 10
 1089 JSR GETDATA

1090 SKIP

1091 PARA
 1092 LMARGIN +17

1093 BLOCK BEGIN

1094 INDENT -19

1095 On

1096 BREAK

1097 Segment number of current segment.

1098 PARA

1099 BLOCK END

1100 BLOCK BEGIN

1101 INDENT -19

1102 On

1103 BREAK

1104 Step number of current step within current segment.

1105 PARA

1106 BLOCK END

1107 BLOCK BEGIN

1108 INDENT -19

1109 On

1110 BREAK

1111 Frequency band code (1 3)

1112 PARA

1113 BLOCK END

1114 BLOCK BEGIN

1115 INDENT -19

1116 An

1117 BREAK

1118 Pointer to buffer to place baseline data.

1119 BLOCK END

1120 LMARGIN RESET

1121 PARA

1122 UNDERSCORE

1123 INDENT -5

1124 5 0 DISC FILE: USED

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: |ALGO.IGR.MC68000_DF|DFALG.FMT NUM

```

1125 .PARA
1126 .UNDERScore
1127 .INDENT -5
1128 6.0 EXTERNAL AND COMMON REFERENCES
1129 .PARA
1130 .UNDERScore
1131 .INDENT -5
1132 7.0 HIGHER REFERENCES
1133 .FILL
1134 .PARA
1135 .LMARGIN +15
1136 .INDENT -15
1137 DFA3.MOD
1138 DF data input.
1139 .LMARGIN RESET
1140 .PARA
1141 .UNDERScore
1142 .INDENT -5
1143 8.0 LOWER REFERENCES
1144 .EJECT
1145 .TITLE?INQUAD.MID? BOTTOM ?PAGE ??
1146 .TITLE?INQUAD.MID?DFALG.NST?PAGE ??
1147 .PARA
1148 .UNDERScore
1149 .INDENT -5
1150 1.0 MODULE NAME
1151 .PARA
1152 .FILL
1153 INQUAD.MOD
1154 .PARA
1155 .UNDERScore
1156 .INDENT -5
1157 2.0 PROGRAMMER
1158 .PARA
1159 .FILL
1160 ETO
1161 .PARA
1162 .UNDERScore
1163 .INDENT -5
1164 3.0 DESCRIPTION
1165 .PARA
1166 .FILL
1167 This module takes the in-phase and quadrature
1168 measurement sums to derive a phase angle for
1169 either the direct or reverse rectifying of the baseline.
1170 The formula to derive a phase angle is
1171 .nofill

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000 DF]DFALG. FMT NUM

```

1172 phase angle=AR.TAN(sum of quad./sum of inphase)
1173 .fill
1174 .PARA
1175 .UNDERSCORE
1176 .INDENT -5
1177 4.0 CALLING SEQUENCE
1178 .PARA
1179 .FILL
1180 .skip
1181 .in 10
1182 JSR INPQUAD
1183 .SKIP
1184 .PARA
1185 .LMARGIN +17
1186 .BLOCK BEGIN
1187 .INDENT -19
1188 Dn
1189 .BREAK
1190 Frequency band code (1-3). Used to determine
1191 number of baselines.
1192 .PARA
1193 .BLOCK END
1194 .BLOCK BEGIN
1195 .INDENT -19
1196 An
1197 .BREAK
1198 Pointer to buffer containing in phase and
1199 quadrature components for all the baselines.
1200 The buffer is set up so that the accumulator
1201 for the inphase sum is followed by the accumulator
1202 for the quadrature sum.
1203 .PARA
1204 .BLOCK END
1205 .BLOCK BEGIN
1206 .INDENT -19
1207 An
1208 .BREAK
1209 Pointer to buffer where phase angles are to
1210 be stored.
1211 .BLOCK END
1212 .LMARGIN RESET
1213 .PARA
1214 .UNDERSCORE
1215 .INDENT -5
1216 5.0 DISC FILE USED
1217 .PARA
1218 .UNDERSCORE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFALG.FMT NUM

```

1219 .INDENT -5
1220 6.0 EXTERNAL AND COMMON REFERENCES
1221 .PARA
1222 .UNDERScore
1223 .INDENT -5
1224 7.0 HIGHER REFERENCES
1225 .FILL
1226 .PARA
1227 .LMARGIN +15
1228 .INDENT -15
1229 DFA3.MOD
1230 DF data input.
1231 .LMARGIN RESET
1232 .PARA
1233 .LMARGIN +15
1234 .INDENT -15
1235 DFA5.MOD
1236 Accumulate data for Fine DF.
1237 .LMARGIN RESET
1238 .PARA
1239 .UNDERScore
1240 .INDENT -5
1241 8.0 LOWER REFERENCES
1242 .FILL
1243 .PARA
1244 .LMARGIN +15
1245 .INDENT -15
1246 ARCTAN.MOD
1247 Arctangent of two values
1248 .LMARGIN RESET
1249 .EJECT
1250 .BTITLE?P2SRCH.M?P2 BOTTOM ?PAGE Z?
1251 .TTITLE?P2SRCH.M?P2DFALG.M?P2PAGE Z?
1252 .PARA
1253 .UNDERScore
1254 .INDENT -5
1255 1.0 MODULE NAME
1256 .PARA
1257 .FILL
1258 P2SRCH.MOD
1259 .PARA
1260 .UNDERScore
1261 .INDENT -5
1262 2.0 PROGRAMMER
1263 .PARA
1264 .FILL
1265 ETO

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFALG.FMT NUM

```

1266 .PARA
1267 .UNDERSCORE
1268 .INDENT -5
1269 3.0 DESCRIPTION
1270 .PARA
1271 .FILL
1272 This module performs the pass 2 search (or fine
1273 search) of the Cal tables during the LOP
1274 calculation. In the second pass, only the
1275 window (+ and - 12 degrees) about the AOA
1276 found in the first pass is examined.
1277 Full interpolation over frequency is done
1278 before comparing the measured phase angles
1279 with the Cal tables.
1280 .PARA
1281 .UNDERSCORE
1282 .INDENT -5
1283 4.0 CALLING SEQUENCE
1284 .PARA
1285 .FILL
1286 .skip
1287 .in 10
1288 JSR P2SRCH
1289 .skip
1290 .PARA
1291 .LMARGIN +17
1292 .BLOCK BEGIN
1293 .INDENT -19
1294 Dn
1295 .BREAK
1296 DF frequency in integer.
1297 .PARA
1298 .BLOCK END
1299 .BLOCK BEGIN
1300 .INDENT -19
1301 Dn
1302 .BREAK
1303 Frequency band code (1-3)
1304 .PARA
1305 .BLOCK END
1306 .BLOCK BEGIN
1307 .INDENT -19
1308 An
1309 .BREAK
1310 Pointer to buffer containing phase angle
1311 measurements for each baseline
1312 .PARA

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFALG.FMT NUM

```

1313 .BLOCK END
1314 .BLOCK BEGIN
1315 .INDENT -19
1316   Dn
1317 .BREAK
1318 Roll angle (BAM) for roll angle compensation
1319 .PARA
1320 .BLOCK END
1321 .BLOCK BEGIN
1322 .INDENT -19
1323   Dn
1324 .BREAK
1325 Index of AOA for search window. Center of window.
1326 .PARA
1327 .BLOCK END
1328 .BLOCK BEGIN
1329 .INDENT -19
1330   Dn
1331 .BREAK
1332 Index of AOA found by second pass search.
1333 .PARA
1334 .BLOCK END
1335 .BLOCK BEGIN
1336 .INDENT -19
1337   Dn
1338 .BREAK
1339 Status of second pass search.
1340 .BLOCK END
1341 .LMARGIN RESET
1342 .PARA
1343 .UNDERScore
1344 .INDENT -5
1345 5.0 DISC FILES USED
1346 .PARA
1347 .UNDERScore
1348 .INDENT -5
1349 6.0 EXTERNAL AND COMMON REFERENCES
1350 .PARA
1351 .UNDERScore
1352 .INDENT -5
1353 7.0 HIGHER REFERENCES
1354 .FILL
1355 .PARA
1356 .LMARGIN +15
1357 .INDENT -15
1358 DFA6.MOD
1359 LOP calculation
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFALG.FMT NUM

```

1360 .LMARGIN RESET
1361 .PARA
1362 .UNDERSCORE
1363 .INDENT -5
1364 8.0 LOWER REFERENCES
1365 .EJECT
1366 .BTITLE?PARFIT.MOD? BOTTOM?PAGE?Z?
1367 .TTITLE?PARFIT.MOD?DFALG.MUT?PAGE?Z?
1368 .PARA
1369 .UNDERSCORE
1370 .INDENT -5
1371 1.0 MODULE NAME
1372 .PARA
1373 .FILL
1374 PARFIT.MOD
1375 .PARA
1376 .UNDERSCORE
1377 .INDENT -5
1378 2.0 PROGRAMMER
1379 .PARA
1380 .FILL
1381 ETO
1382 .PARA
1383 .UNDERSCORE
1384 .INDENT -5
1385 3.0 DESCRIPTION
1386 .PARA
1387 .FILL
1388 This module interpolates the index of the AOA determined
1389 by the second pass search of the LOP calculation.
1390 The interpolation is performed as a parabolic fit of the
1391 index and the two adjacent points. The calculated vertex
1392 of the parabola is the AOA for the DF.
1393 .PARA
1394 .UNDERSCORE
1395 .INDENT -5
1396 4.0 CALLING SEQUENCE
1397 .PARA
1398 .FILL
1399 .skip
1400 .in 10
1401 JSR PARFIT
1402 .skip
1403 .PARA
1404 .LMARGIN +17
1405 .BLOCK BEGIN
1406 .INDENT -19

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000_DF]DFALG.FMT NUM

1407	Dn	Int.*2/Input
1408	.BREAK	
1409	Index of AOA found in second pass search	
1410	.PARA	
1411	.BLOCK END	
1412	.BLOCK BEGIN	
1413	.INDENT -19	
1414	Dn	Int.*2/Output
1415	.BREAK	
1416	Interpolated AOA for DF	
1417	.BLOCK END	
1418	.LMARGIN RESET	
1419	.PARA	
1420	.UNDERScore	
1421	.INDENT -5	
1422	5.0 DISC FILE USED	
1423	.PARA	
1424	.UNDERScore	
1425	.INDENT -5	
1426	6.0 EXTERNAL AND COMMON REFERENCES	
1427	.PARA	
1428	.UNDERScore	
1429	.INDENT -5	
1430	7.0 HIGHER REFERENCES	
1431	.FILL	
1432	.PARA	
1433	.LMARGIN +15	
1434	.INDENT -15	
1435	DFA6.MOD	
1436	LOP calculation	
1437	.LMARGIN RESET	
1438	.PARA	
1439	.UNDERScore	
1440	.INDENT -5	
1441	8.0 LOWER REFERENCES	
1442	.EJECT	
1443	.TITLE?READACC.MID? BOTTOM ?PAGE ??	
1444	.TITLE?READACC.MID?DFALG.MST?PAGE ??	
1445	.PARA	
1446	.UNDERScore	
1447	.INDENT -5	
1448	1.0 MODULE NAME	
1449	.PARA	
1450	.FILL	
1451	READACC.MOD	
1452	.PARA	
1453	.UNDERScore	

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO,IGR,MC68000_DF]DFALG.FMT_NUM

```

1454 .INDENT -5
1455 2.0 PROGRAMMER
1456 .PARA
1457 .FILL
1458 ETO
1459 .PARA
1460 .UNDERScore
1461 .INDENT -5
1462 3.0 DESCRIPTION
1463 .PARA
1464 .FILL
1465 This module read: the AGC value from the DF Dual
1466 channel receiver. The appropriate commands are
1467 sent to the IF processor (within the Fast DF
1468 Controller) to take a reading of the AGC value.
1469 .PARA
1470 .UNDERScore
1471 .INDENT -5
1472 4.0 CALLING SEQUENCE
1473 .PARA
1474 .FILL
1475 .skip
1476 .in 10
1477 JSR READAGC
1478 .SKIP
1479 .PARA
1480 .LMARGIN +17
1481 .BLOCK BEGIN
1482 .INDENT -19
1483 Dn
1484 .BREAK
1485 AGC value reading.
1486 .BLOCK END
1487 .LMARGIN RESET
1488 .PARA
1489 .UNDERScore
1490 .INDENT -5
1491 5.0 DISC FILE USED
1492 .PARA
1493 .UNDERScore
1494 .INDENT -5
1495 6.0 EXTERNAL AND COMMON REFERENCES
1496 .PARA
1497 .UNDERScore
1498 .INDENT -5
1499 7.0 HIGHER REFERENCES
1500 .FILL
  
```

Int*4/Output

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFALG.FMT NUM

```

1501 .PARA
1502 .LMARGIN +15
1503 .INDENT -15
1504 DFA3.MOD
1505 DF data input.
1506 .LMARGIN RESET
1507 .PARA
1508 .UNDERScore
1509 .INDENT -5
1510 8.0 LOWER REFERENCES
1511 .EJECT
1512 .TITLE?SETUP.MOD? BOTTOM ?PAGE ??
1513 .TITLE?SETUP.MOD?DFALG.MST?PAGE ??
1514 .PARA
1515 .UNDERScore
1516 .INDENT -5
1517 1.0 MODULE NAME
1518 .PARA
1519 .FILL
1520 SETUP.MOD
1521 .PARA
1522 .UNDERScore
1523 .INDENT -5
1524 2.0 PROGRAMMER
1525 .PARA
1526 .FILL
1527 ETO
1528 .PARA
1529 .UNDERScore
1530 .INDENT -5
1531 3.0 DESCRIPTION
1532 .PARA
1533 .FILL
1534 This job sets up the commands to the various pieces of
1535 hardware (DF receiver, RF processor, Antenna Control units)
1536 for the next step of the current DF segment. Based on step
1537 and segment, the commands for the next data collection
1538 sequence are loaded into the appropriate output interfaces
1539 (SIO or Parallel output). The number of baselines and
1540 which baselines is also controlled by frequency band.
1541 .PARA
1542 .UNDERScore
1543 .INDENT -5
1544 4.0 CALLING SEQUENCE
1545 .PARA
1546 .FILL
1547 .skip

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1: [ALGO: IGR.MC68000_DF]DFALG.FMT NUM

```

1548 .in 10
1549 JSR SETUP
1550 .skip
1551 .PARA
1552 .LMARGIN +17
1553 .BLOCK BEGIN
1554 .INDENT -19
1555 .Dn
1556 .BREAK
1557 .Frequency band code (1-3)
1558 .PARA
1559 .BLOCK END
1560 .BLOCK BEGIN
1561 .INDENT -19
1562 .Dn
1563 .BREAK
1564 .Segment number for current DF.
1565 .PARA
1566 .BLOCK END
1567 .BLOCK BEGIN
1568 .INDENT -19
1569 .Dn
1570 .BREAK
1571 .Step number of next data collection sequence.
1572 .BLOCK END
1573 .LMARGIN RESET
1574 .PARA
1575 .UNDERSCORE
1576 .INDENT -5
1577 5.0 DISC FILES USED
1578 .PARA
1579 .UNDERSCORE
1580 .INDENT -5
1581 6.0 EXTERNAL AND COMMON REFERENCES
1582 .PARA
1583 .UNDERSCORE
1584 .INDENT -5
1585 7.0 HIGHER REFERENCES
1586 .FILL
1587 .PARA
1588 .LMARGIN +15
1589 .INDENT -15
1590 DFA2.MPD
1591 Start/continue DF data collection
1592 .LMARGIN RESET
1593 .PARA
1594 .UNDERSCORE
  
```

IMPROVED GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFALG.FMT NUM

```

1595 .INDENT -5
1596 8.0 LOWER REFERENCES
1597 .EJECT
1598 .BTITLE?STRD.F.M.M? BOTTOM ?PAGE ??
1599 .TTITLE?STRD.F.M.M?DFALG.MST?PAGE ??
1600 .PARA
1601 .UNDERSORE
1602 .INDENT -5
1603 1.0 MODULE NAME
1604 .PARA
1605 .FILL
1606 STRD.F.MOD
1607 .PARA
1608 .UNDERSORE
1609 .INDENT -5
1610 2.0 PROGRAMMER
1611 .PARA
1612 .FILL
1613 ETO
1614 .PARA
1615 .UNDERSORE
1616 .INDENT -5
1617 3.0 DESCRIPTION
1618 .PARA
1619 .FILL
1620 This module sends the appropriate command to the
1621 Fast Df Controller to initiate the current Df
1622 data collection sequence. The correct command is
1623 determined based on frequency band, segment number,
1624 and step of the current Df.
1625 The command to the Fast Df controller controls
1626 how many commands and to which pieces of hardware
1627 the commands are to be sent to perform the measurement.
1628 .PARA
1629 .UNDERSORE
1630 .INDENT -5
1631 4.0 CALLING SEQUENCE
1632 .PARA
1633 .FILL
1634 .SKIP
1635 in 10
1636 JSR STRD.F
1637 .SKIP
1638 .PARA
1639 LMARGIN +17
1640 .BLOCK BEGIN
1641 .INDENT -19

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000, DF]DFALG FMT NUM

```

1642 Dn Int*2/Input.
1643 .BREAK
1644 Frequency band code (1-3)
1645 .PARA
1646 .BLOCK END
1647 .BLOCK BEGIN
1648 .INDENT -19
1649 Dn Int*2/Input.
1650 .BREAK
1651 Segment number of current DF.
1652 .PARA
1653 .BLOCK END
1654 .BLOCK BEGIN
1655 .INDENT -19
1656 Dn Int*2/Input.
1657 .BREAK
1658 Step number of current data collection sequence.
1659 .BLOCK END
1660 .LMARGIN RESET
1661 .PARA
1662 .UNDERScore
1663 .INDENT -5
1664 5.0 DISC FILE: USED
1665 .PARA
1666 .UNDERScore
1667 .INDENT -5
1668 6.0 EXTERNAL AND COMMON REFERENCES
1669 .PARA
1670 .UNDERScore
1671 .INDENT -5
1672 7.0 HIGHER REFERENCES
1673 .FILL
1674 .PARA
1675 .LMARGIN +15
1676 .INDENT -15
1677 DFAZ,MOD
1678 Start/continue DF data collection
1679 .LMARGIN RESET
1680 .PARA
1681 .UNDERScore
1682 .INDENT -5
1683 8.0 LOWER REFERENCES
1684 .EJECT

```

Improved GUARDRAIL V MC68000 'DF' Files
 DFG1:[ALGO.IGR.MC68000_DF]DFG1.FMT NUM

16 8 DFG1.FMT_NUM

***** Source Listing --> DFG1.FMT_NUM *****

LFEN 116

MODULE NAME

DFG1 MOD

PROGRAM# R

P. Eto

DESCRIPTION

DFG1

DF requests can originate from the Perkin Elmer (manual, BITE, Interoperable, calibration), the AOI function, the Geoscreen function, Auto DF, or the General Search process. The DF request input is scheduled from any of these request originators with a similar request message which contains codes to identify the type (BITE, manual, Auto DF, General Search, Geoscreen, AOI, Interoperable, or Calibration), and subtype (coarse, fine). These codes are extracted from the message along with the DF parameters, and are placed in the appropriate request queue. After the DF process has been initiated, this job terminates. When starting the DF process, this job will schedule the initial call to the Remove old DF request job (see DFG2).

CALLING SEQUENCE

?SCHED DFG1.#4,,,IPACK

IPACK

Int*4/Input
 Packet pointer to DF request. Packet
 definition "DF" or "DG".

DISC FILE USED

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL [ALGO. IGR. MC68000_DF]DFGL.FMT NUM

EXTERNAL AND CONTROL REFERENCES

HIGHER REFERENCES

DFPRT.MOD	Receive DF requests from the Perkin Elmer
ADFRST.MOD	Check for and auto DF request and send it
AOIIP.MOD	Area of Interest Analysis
SCAR NE.MOD	Performs the SCAR 1 analysis
SCAR WO.MOD	This routine performs SCAR 2 analysis
ARFDBIT.MOD	ARF DF Bite Request Job
GEORST.MOD	Georeferenced DF Request Job
GEODL.MOD	Georeferencing Analysis Routine
ESGR.MOD	Receives directed search reports from the ADP

LOWREFERENCES

GETN.MOD	Determine number of DF requests
GETP.MOD	Get pointer to available element of DF queue
REF.MOD	Remove old DF requests
REF.MOD	Remove DF request (from bottom)
REF.MOD	Assign the next account number
REF.MOD	Add request to DF request queue
REF.MOD	Use a small packet for reuse

AD-A184 057

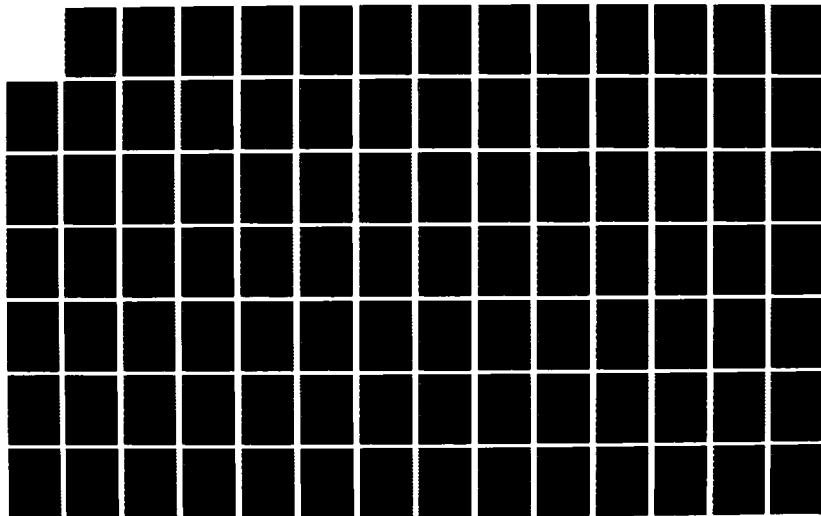
US ARMY INTELLIGENCE CENTER AND SCHOOL (USAICS)
SOFTWARE ANALYSIS AND MAN (U) JET PROPULSION LAB
PASADENA CA B PARD0 85 MAR 87 JPL-D-4216 NAS7-918

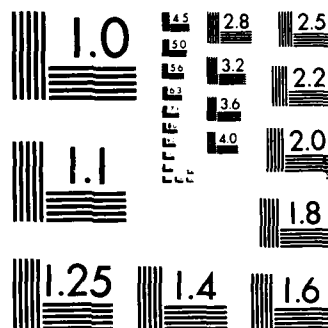
3/5

UNCLASSIFIED

F/G 12/5

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

16.9 DFG2.FMT_NUM

***** Source Listing --> DFG2.FMT_NUM *****

1 LLEN 116

2 MODULE NAME

4 DFG2.MOD

7 PROGRAMMER

9 P. Eto

12 DESCRIPTION

17 DFG2

This job is initially scheduled by the DF Request Input Job (DFG1) to screen the DF request queues for old requests. The DF request "queues" are not true queues. As requests are input (by the DF Request Input DFG1), the requests are "pushed" onto the queue, and when requests are processed, the requests are "popped" so that the most recent requests are processed. Hence the "queues" are more like stacks in the normal use. To remove old requests, however, the search starts at the oldest requests (the "bottom") and works toward the more recent requests. The old requests are removed in this manner, using the structure like a true queue. When a request is removed, a message indicating the rejection is sent to the originating function (Perkin Elmer, Geoscreen, AOI, Auto DF, General Search). After the initial call, this job reschedules itself after a wait of a set amount of time (tbd).

CALLING SEQUENCE

?SCHED DFG2,#4,...

DISC FILES USED

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG2.FMT_NUM

44 *
 45 *
 46 *
 47 *
 48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *
 59 *
 60 *
 61 *
 62 *
 63 *
 64 *

EXTERNAL AND COMMON REFERENCES

HB Database
 Header block data base for DF queues.

HIGHER REFERENCES

DFG2.MOD Remove old DF requests.
 DFG1.MOD DF request input.

LOWERREFERENCES

DFG2.MOD Remove old DF requests.
 DFPF2.MOD Report DF to Perkin Elmer.
 FINOLD.MOD Find and remove old DF requests.

DFG3.FMT NUM

16.10

```
***** Source Listing ==> DFG3.FMT_NUM *****
```

1	LLN 116
2	MODULE NAME

DEGR MOD

PROGRAMME R

015

DESCRIPTION

DFG3

After the DF process is initiated, the DF Scheduler is called every time a SCAR 1 request is transmitted to the ADPUs. The DF Scheduler determines the next DF segment to be processed. The DF Scheduler determines when to interleave coarse and fine DF segments as well as priority among fine requests. Once a DF segment is determined (fine or coarse), the segment is examined to determine if a response will be generated. If a response will be generated, a control packet is created, and the DF control packet (which contains the link to the DF request message) is placed on the DF output queue. Otherwise, just the DF request is placed on the DF output queue. On the next output for a DF request, the Uplink Handler will use the request in the DF output queue. If the DF request requires Audio Correlation, this job will check the Audio Correlator ready flag to determine if the Audio Correlator is free. If it is, the request for Audio Correlation will be made. If the DF request is for calibration, the signal generator will be requested by setting a flag in local common. In the case of Interop. DF requests, a flag is set to indicate an Interop. DF is being processed. The DF scheduler is called by the report to Parkin Elmer job (DFPE2) when the Interop. DF completes.

CALLING SEQUENCE

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO. IGR. MC68000_DF]DFG3.FMT NUM

44 *
 45 *
 46 *
 47 *
 48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *
 59 *
 60 *
 61 *
 62 *
 63 *
 64 *
 65 *
 66 *
 67 *
 68 *
 69 *
 70 *
 71 *
 72 *
 73 *

JSR DFG3

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

GAQUEUER.MOD Prepare next message for sending up link.

DFPE2.MOD Report DF to Perkin Elmer.

LOWERREFERENCES

GETCRS.MOD Get next coarse DF request.

GETFINE.MOD Get next fine DF request.

GETBITS.MOD Access BITS fields.

PUTBITS.MOD Modify BITS fields.

GETPAK.MOD Allocate a small packet.

GASEND.MOD Queue messages to be sent up link.

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG5.FMT NUM

DFG5.FMT_NUM

16.11

***** Source Listing -> DFG5.FMT_NUM *****

```

1  LLEN 116
2  MODULE NAME
3
4  DFG5 MOD
5
6
7  PROGRAMMER
8
9  SMAN ION
10
11
12  DESCRIPTION
13
14  This job controls the signal generator for
15  calibration/verification and BITE DF requests.
16
17  CALLING SEQUENCE
18
19
20
21  ?SCHED DFG5,02,...,IPACK
22
23
24  IPACK      Int*4/Inp't.
25            Packet pointer to signal generator request.
26            Packet definition "DFP".
27
28
29  DISC FILES USED
30
31
32  EXTERNAL AND COMMON REFERENCES
33
34
35  HIGHER REFERENCES
36
37  DFG6 MOD      DF immediate job.
38
39
40  LOWERREFERENCES
41
42  RELPAK MOD      Release any valid combination of packets.
43

```

Improved GUARDRAIL V MC68000 'DF' Files
DRCL: [ALGO.IGR.MC68000_DF]DFG5.FMT NUM

44 *	BCD-IN.M00	Convert BCD to binary integer.
45 *		
46 *	DFG-T.M00	Signal Generator Timeout

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFG5T.FMT_NUM

16.12 DFG5T.FMT_NUM

***** Source Listing ==> DFG5T.FMT_NUM *****

 LLEN 116

MODULE NAME

DFG5T.MOD

PROGRAMMER

SWANSON

DESCRIPTION

This job turns off the Signal Generator if it has not been
 returned within the specified timeout length.

CALLING SEQUENCE

?SCHFD DFG5T,#4,#100,#1

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

DFG5T.MOD Signal generator control.

DFG5T.MOD Signal Generator timeout

LOWERREFERENCES

DFG5T.MOD Signal Generator timeout

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG6.FMT_NUM

16.13 DFG6.FMT_NUM

***** Source Listing --> DFG6.FMT_NUM *****

1 LLEN 116

2 MODULE NAME

3 DFG6.MOD

4 PROGRAMMER

5 SHANTON

6 DESCRIPTION

7 DFG6

This job is scheduled by the Uplink dequeuer (GAQUEUER) as the immediate job. This job is scheduled when a SCAR 2 request is transmitted up the link. If the audio correlator is requested, this job sets up a timeout for the Audio Correlator control job (DFG8), and sets the Audio Correlator flag to indicate that the Audio Correlator is unavailable. If the signal generator has been requested, this job will set up timeout for the Signal Generator control job (DFG5). The timeouts for the signal generator and the audio correlator are to allow for the link transmission time to synchronize the process on the ground with the process in the air.

CALLING SEQUENCE

?SCHED DFG6.#2...

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

SIGREQ Flag

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFG6.FMT_NUM

44 *		Signal Generator request flag
45 *		
46 *	ACORREQ	Flag
47 *		Audio Correlator request flag.
48 *		
49 *	ACORAV	Flag
50 *		Audio Correlator availability.
51 *		
52 *		Database
53 *	CURDF	Current DF frequency.
54 *		
55 *		
56 *		
57 *		
58 *		
59 *	GAQUEUER.MOD	Prepare next message for sending up link.
60 *		
61 *		
62 *		
63 *	GETPAK.MOD	Allocate a small packet.
64 *		
65 *	DFG5.MOD	Signal generator control.
66 *		
67 *	DFG8.MOD	Audio correlator control.

HIGHER REFERENCES

LOWERREFERENCES

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG7.FMT_NUM

16.14 DFG7.FMT_NUM

***** Source Listing -> DFG7.FMT_NUM *****

LEEN 116

MODULE NAME

DFG: MOD

PROGRAMMER

ETO

DESCRIPTION

DFG/

This job is scheduled after the wait (timeout) set up by the start audio correlation job (DFG8). If the current correlation has not completed (reflected in the audio correlator ready flag), then the audio correlator is assumed to have timed out. A message is sent to either the report to PE job (DFP22) or the audio correlator BITE results job (ACF1TR) depending on what the correlation was requested for. If the correlation timed out, the audio correlator ready flag is set (to true).

CALLING SEQUENCE

?SCHED DFG7, #4,...

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

Improved GUARDRAIL V MC68000 'DF' Files
DRCl: [ALGO.IGR.MC68000_DF]DFG7.FMT NUM

44 *
45 *
46 *
47 *

LOWERREFERENCES

GETAK.MOD

Allocate a small packet.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG8.FMT_NUM

DFG8.FMT_NUM

16.15

***** Source Listing --> DFG8.FMT_NUM *****

LEN 116

MODULE NAME

DFG8.MOD

PROGRAMMER

M. Swanson

DESCRIPTION

DFG8

If the next DF request requires Audio correlation, this job is scheduled after a time out by the DF Immediate (DFG6). When the next DF is about to start, this job will command the Audio Correlator to start correlating. When the correlation is complete, the Audio Correlator will interrupt the Audio Correlator input job (see DFG9). The Audio Correlator receives inputs from the Audio Correlation Receiver in each ARF. From the DF request identification message passed to it, this job updates the current audio correlation request parameters (in local common). This job is also scheduled by the Audio Correlator BITE job (ACBITE) to perform the Audio Correlator BITE test. The current Audio Correlator request parameters will reflect the fact that it is a BITE request.

CALLING SEQUENCE

?SCHED DFG8,#2,...,IPACK

IPACK Int*4/Input
 Pointer to packet containing Audio
 Correlator request. Packet definition
 "ACCR".

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFG8.FMT_NUM

44 *
 45 *
 46 *
 47 *
 48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *
 59 *
 60 *
 61 *
 62 *
 63 *
 64 *
 65 *
 66 *
 67 *

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

ACORAV Flag
 Audio Correlator availability.

HIGHER REFERENCES

ACBITE.MOD Audio Correlator Bite Job
 DFG6.MOD DF immediate job.

LOWERREFERENCES

RELPAK.MOD Release any valid combination of packets.
 GETPAK.MOD Allocate a small packet.
 DFPE2.MOD Report DF to Perkin Elmer.
 ACBITR.MOD Audio Correlator Bite Results Job

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFG9.FMT_NUM

DFG9.FMT_NUM

16.16

***** Source Listing ==> DFG9.FMT_NUM *****

 LLEN 116
 MODULE NAME

DFG9.MOD

PROGRAMMER

P. Eto

DESCRIPTION

DFG9

When the Audio Correlator is finished correlating on the current frequency, it will interrupt the GPU and this job will be scheduled. This job will input whether the correlation passed or failed, and relay the result to the Report to the Perkin Elmer job (see DFPE2). This job will clear the Audio Correlation ready flag to indicate that the Audio Correlator is free. If the current correlation is for a RITE request (reflected in the current Audio Correlator request parameters), the correlation results will be sent to the Audio Correlation BITE response job (ACBTR).

CALLING SEQUENCE

?SCHED DFG9,#4,,,

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

1
 2 *
 3 *
 4 *
 5 *
 6 *
 7 *
 8 *
 9 *
 10 *
 11 *
 12 *
 13 *
 14 *
 15 *
 16 *
 17 *
 18 *
 19 *
 20 *
 21 *
 22 *
 23 *
 24 *
 25 *
 26 *
 27 *
 28 *
 29 *
 30 *
 31 *
 32 *
 33 *
 34 *
 35 *
 36 *
 37 *
 38 *
 39 *
 40 *
 41 *
 42 *
 43 *

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1: [ALGO.IGR.MC68000_DF]DFG9.FMT_NUM

44 *
 45 *
 46 *
 47 *
 48 *

LOWERREFERENCES

ACB:TR.MOD

Audio Correlator Bite Results Job

GET:AK.MOD

Allocate a small packet.

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000_DF]DFGRND.FMT_NUM

16.17 DFGRND.FMT_NUM

```
***** Source Listing --> DFGRND.FMT_NUM *****
*****
1 .EJECT
2 .BTITLE?DFG1.MOD? BOTTOM ?PAGE ??
3 .TTITLE?DFG1.MOD?DFGRND.MST?PAGE ??
4 .PARA
5 .UNDERSORE
6 .INDENT -5
7 1.0 MODULE NAME
8 .PARA
9 .FILL
10 DFG1.MOD
11 .PARA
12 .UNDERSORE
13 .INDENT -5
14 2.0 PROGRAMMER
15 .PARA
16 .FILL
17 ETO
18 .PARA
19 .UNDERSORE
20 .INDENT -5
21 3.0 DESCRIPTION
22 .PARA
23 .FILL
24 .imbed DFG1.STR/G
25 .PARA
26 .UNDERSORE
27 .INDENT -5
28 4.0 CALLING SEQUENCE
29 .PARA
30 .FILL
31 .skip
32 .in 10
33 ?SCHED DFG1,#4,...IPACK
34 .SKIP
35 .PARA
36 .LMARGIN +17
37 .BLOCK BEGIN
38 .INDENT -19
39 IPACK
40 .BREAK
41 Packet pointer to DF request. Packet definition "DF" or "DG".
42 .skip
43 .BLOCK END
```

Int.*4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFGRND.FMT NUM

```

44 .LMARGIN RESET
45 .PARA
46 .UNDERScore
47 .INDENT -5
48 5.0 DISC FILE USED
49 .PARA
50 .UNDERScore
51 .INDENT -5
52 6.0 EXTERNAL AND COMMON REFERENCES
53 .PARA
54 .UNDERScore
55 .INDENT -5
56 7.0 HIGHER REFERENCES
57 .PARA
58 .UNDERScore
59 .INDENT -5
60 8.0 LOWER REFERENCES
61 .FILL
62 .PARA
63 .LMARGIN +15
64 .INDENT -15
65 GETNUM.MOD
66 Determine number of DF requests.
67 .LMARGIN RESET
68 .PARA
69 .LMARGIN +15
70 .INDENT -15
71 GETPTR.MOD
72 Get pointer to available element of DF queue.
73 .LMARGIN RESET
74 .PARA
75 .LMARGIN +15
76 .INDENT -15
77 REMBOT.MOD
78 Remove DF request (from bottom).
79 .LMARGIN RESET
80 .PARA
81 .LMARGIN +15
82 .INDENT -15
83 ACCOUNT.MOD
84 Assign the next account number.
85 .LMARGIN RESET
86 .PARA
87 .LMARGIN +15
88 .INDENT -15
89 PUSH MOD
90 Add DF request to DF request queue.

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFGRND.FMT NUM

```

91 .LMARGIN RESET
92 .PARA
93 .LMARGIN +15
94 .INDENT -15
95 PUTPAK.MOD
96 Re'ase a small packet for reuse.
97 .LMARGIN RESET
98 .EJECT
99 .BTITLE?DFG2.MOD? BOTTOM ?PAGE ??
100 .TTITLE?DFG2.MOD?DFGRND.MST?PAGE ??
101 .PARA
102 .UNDERScore
103 .INDENT -5
104 1.0 MODULE NAME
105 .PARA
106 .FILL
107 DFG2.MOD
108 .PARA
109 .UNDERScore
110 .INDENT -5
111 2.0 PROGRAMMER
112 .PARA
113 .FILL
114 ETO
115 .PARA
116 .UNDERScore
117 .INDENT -5
118 3.0 DESCRIPTION
119 .PARA
120 .FILL
121 .imbed DFG2.STR/G
122 .PARA
123 .UNDERScore
124 .INDENT -5
125 4.0 CALLING SEQUENCE
126 .PARA
127 .FILL
128 .skip
129 .in 10
130 ?SCHED DFG2.#4...
131 .skip
132 .FILL
133 .PARA
134 .UNDERScore
135 .INDENT -5
136 5.0 DISC FILE: USED
137 .PARA

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

138 .UNDERSCORE
139 .INDENT -5
140 6.0 EXTERNAL AND COMMON REFERENCES
141 .PARA
142 .FILL
143 .LMARGIN +17
144 .BLOCK BEGIN
145 .INDENT -19
146 HB Database
147 .BREAK
148 Header block data base for DF queues.
149 .BLOCK END
150 .LMARGIN RESET
151 .PARA
152 .UNDERSCORE
153 .INDENT -5
154 7.0 HIGHER REFERENCES
155 .PARA
156 .UNDERSCORE
157 .INDENT -5
158 8.0 LOWER REFERENCES
159 .FILL
160 .PARA
161 .LMARGIN +15
162 .INDENT -15
163 FINDOLD.MOD
164 Find and remove old DF requests.
165 .LMARGIN RESET
166 .EJECT
167 .BTITLE?DFG3.MOD? BOTTOM ?PAGE ??
168 .TTITLE?DFG3.MOD?DFGRND.MST?PAGE ??
169 .PARA
170 .UNDERSCORE
171 .INDENT -5
172 1.0 MODULE NAME
173 .PARA
174 .FILL
175 DFG3.MOD
176 .PARA
177 .UNDERSCORE
178 .INDENT -5
179 2.0 PROGRAMMER
180 .PARA
181 .FILL
182 ETO
183 .PARA
184 .UNDERSCORE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO:IGR.MC68000_DF]DFGRND.FMT_NUM

```

185 .INDENT -5
186 3 0 .DESCRIPTION
187 . PARA
188 . FILL
189 .inbed DPG3 STR/
190 . PARA
191 . UNDERSCORE
192 .INDENT -5
193 4 0 .CALLING SEQUENCE
194 . PARA
195 . FILL
196 . skip
197 .in 10
198 .PSCHED DPG3.0 ...
199 . skip
200 . FILL
201 . PARA
202 . UNDERSCORE
203 .INDENT -5
204 5 0 .DISC FILF USED
205 . PARA
206 . UNDERSCORE
207 .INDENT -5
208 6 0 .EXTERNAL AND COMMON REFERENCES
209 . PARA
210 . UNDERSCORE
211 .INDENT -5
212 7 0 .HIGHER REFERENCES
213 . PARA
214 . UNDERSCORE
215 .INDENT -5
216 8 0 .LOWER REFERENCES
217 . FILL
218 . PARA
219 .LMARGIN +15
220 .INDENT -15
221 .GETCRS MOD
222 .Get next coarse IF request
223 .LMARGIN RESET
224 . PARA
225 .LMARGIN +15
226 .INDENT -15
227 .GETFINE MOD
228 .Get next fine DF request
229 .LMARGIN RESET
230 .EJECT
231 .BTITLE'DPG5.MOD' BOTTOM ?PAGE ??

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

232 TITLE'DFGS MOD 'PGRND MCT?PAGE 1?
233 PARA
234 UNDERSCORE
235 INDENT -5
236 1.0 MODULE NAME
237 PARA
238 FILL
239 DFGS MOD
240 PARA
241 UNDERSCORE
242 INDENT -5
243 2.0 PROGRAMMER
244 PARA
245 FILL
246 ETO
247 PARA
248 UNDERSCORE
249 INDENT -5
250 3.0 DESCRIPTION
251 PARA
252 FILL
253 This job controls the signal generator for
254 calibration/verification and BITE DF requests.
255 PARA
256 UNDERSCORE
257 INDENT -5
258 4.0 CALLING SEQUENCE
259 PARA
260 FILL
261 skip
262 in 10
263 ?SCHED DFGS.#2,...IPACK
264 SKIP
265 PARA
266 LMARGIN +17
267 BLOCK BEGIN
268 INDENT -19
269 IPACK
270 BREAK
271 Packet pointer to signal generator request. Packet
272 definition "DFP"
273 BLOCK END
274 LMARGIN RESET
275 PARA
276 UNDERSCORE
277 INDENT -5
278 5.0 DISC FILE USED
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFGRND.FMT_NUM

```

279 . PARA
280 . UNDERSCORE
281 . INDENT -5
282 6 0 . EXTERNAL AND COMMON REFERENCES
283 . PARA
284 . UNDERSCORE
285 . INDENT -5
286 7 0 . HIGHER REFERENCES
287 . PARA
288 . UNDERSCORE
289 . INDENT -5
290 8 0 . LOWER REFERENCES
291 . FILL
292 . PARA
293 . LMARGIN +15
294 . INDENT -15
295 PUTPAK MOD
296 Release a small packet for reu...
297 . LMARGIN RESET
298 . EJECT
299 .BTITLE?DFG6 MOD . BOTTOM ?PAGE ??
300 .TTITLE?DFG6 MOD .DFGRND.MST?PAGE ??
301 . PARA
302 . UNDERSCORE
303 . INDENT -5
304 1 0 . MODULE NAME
305 . PARA
306 . FILL
307 DFG6 MOD
308 . PARA
309 . UNDERSCORE
310 . INDENT -5
311 2 0 . PROGRAMMER
312 . PARA
313 . FILL
314 ETO
315 . PARA
316 . UNDERSCORE
317 . INDENT -5
318 3 0 . DESCRIPTION
319 . PARA
320 . FILL
321 .inbed DFG6 STR/C
322 . PARA
323 . UNDERSCORE
324 . INDENT -5
325 4 0 . CALLING SENTENCE

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFGRND.FMT NUM

```

326 . PARA
327 . FILL
328 . skip
329 . in 10
330 ?SCHED DFG6.#2...
331 . SKIP
332 . FILL
333 . PARA
334 . UNDERSORE
335 . INDENT -5
336 5.0 DISC FILE USED
337 . PARA
338 . UNDERSORE
339 . INDENT -5
340 6.0 EXTERNAL AND COMMON REFERENCES
341 . PARA
342 . UNDERSORE
343 . INDENT -5
344 7.0 HIGHER REFERENCES
345 . PARA
346 . UNDERSORE
347 . INDENT -5
348 8.0 LOWER REFERENCES
349 . FILL
350 . PARA
351 .LMARGIN +15
352 .INDENT -15
353 GETPAK.MOD
354 Allocate a small packet.
355 .LMARGIN RESET
356 .EJECT
357 .BTITLE?DFG7.MOD? BOTTOM ?PAGE ??
358 .TTITLE?DFG7.MOD?DFGRND.MST?PAGE ??
359 . PARA
360 . UNDERSORE
361 . INDENT -5
362 1.0 MODULE NAME
363 . PARA
364 . FILL
365 DFG7.MOD
366 . PARA
367 . UNDERSORE
368 . INDENT -5
369 2.0 PROGRAMMER
370 . PARA
371 . FILL
372 ETO

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

373 . PARA
374 . UNDERSORE
375 . INDENT -5
376 3.0 DESCRIPTION
377 . PARA
378 . FILL
379 . imbed DFG7.STR/G
380 . PARA
381 . UNDERSORE
382 . INDENT -5
383 4.0 CALLING SEQUENCE
384 . PARA
385 . FILL
386 . skip
387 . in 10
388 ?SCHED DFG7.#4,...
389 . skip
390 . FILL
391 . PARA
392 . UNDERSORE
393 . INDENT -5
394 5.0 DISC FILES USED
395 . PARA
396 . UNDERSORE
397 . INDENT -5
398 6.0 EXTERNAL AND COMMON REFERENCES
399 . PARA
400 . UNDERSORE
401 . INDENT -5
402 7.0 HIGHER REFERENCES
403 . PARA
404 . UNDERSORE
405 . INDENT -5
406 8.0 LOWER REFERENCES
407 . FILL
408 . PARA
409 .LMARGIN +15
410 . INDENT -15
411 GETPAK.MOD
412 Allocate a small packet.
413 .LMARGIN RESET
414 .EJECT
415 .BTITLE?DFG8.MOD? BOTTOM ?PAGE ??
416 .TTITLE?DFG8.MOD?DFGRND.MST?PAGE.??
417 . PARA
418 . UNDERSORE
419 . INDENT -5

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

420 1.0  MODULE NAME
421 .PARA
422 .FILL
423 DFG8.MOD
424 .PARA
425 .UNDERScore
426 .INDENT -5
427 2.0  PROGRAMMER
428 .PARA
429 .FILL
430 ETO
431 .PARA
432 .UNDERScore
433 .INDENT -5
434 3.0  DESCRIPTION
435 .PARA
436 .FILL
437 .imbed DFG8.STR/C
438 .PARA
439 .UNDERScore
440 .INDENT -5
441 4.0  CALLING SEQUENCE
442 .PARA
443 .FILL
444 .skip
445 .in 10
446 ?SCHED DFG8.#2...IPACK
447 .SKIP
448 .PARA
449 .LMARGIN +17
450 .BLOCK BEGIN
451 .INDENT -19
452 IPACK
453 .BREAK
454 Pointer to packet containing Audio Correlator
455 request. Packet definition "ACCR".
456 .BLOCK END
457 .LMARGIN RESET
458 .PARA
459 .UNDERScore
460 .INDENT -5
461 5.0  DISC FILES USED
462 .PARA
463 .UNDERScore
464 .INDENT -5
465 6.0  EXTERNAL AND COMMON REFERENCES
466 .PARA
  
```

Int**4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

467 .UNDERSCORE
468 .INDENT -5
469 7.0 HIGHER REFERENCES
470 .PARA
471 .UNDERSCORE
472 .INDENT -5
473 8.0 LOWER REFERENCES
474 .FILL
475 .PARA
476 .LMARGIN +15
477 .INDENT -15
478 PUTPAK.MOD
479 Release a small packet for reuse.
480 .LMARGIN RESET
481 .EJECT
482 .BTITLE?DFG9.MOD? BOTTOM ?PAGE 1?
483 .TTITLE?DFG9.MOD?DFGRND.MST?PAGE 1?
484 .PARA
485 .UNDERSCORE
486 .INDENT -5
487 1.0 MODULE NAME
488 .PARA
489 .FILL
490 DFG9.MOD
491 .PARA
492 .UNDERSCORE
493 .INDENT -5
494 2.0 PROGRAMMER
495 .PARA
496 .FILL
497 ETO
498 .PARA
499 .UNDERSCORE
500 .INDENT -5
501 3.0 DESCRIPTION
502 .PARA
503 .FILL
504 .imbed DFG9.STR/G
505 .PARA
506 .UNDERSCORE
507 .INDENT -5
508 4.0 CALLING SEQUENCE
509 .PARA
510 .FILL
511 .skip
512 .in 10
513 ?SCHED DFG9.#A...

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

514 .SKIP
515 .FILL
516 .PARA
517 .UNDERScore
518 .INDENT -5
519 5.0 DISC FILE: USED
520 .PARA
521 .UNDERScore
522 .INDENT -5
523 6.0 EXTERNAL AND COMMON REFERENCES
524 .PARA
525 .UNDERScore
526 .INDENT -5
527 7.0 HIGHER REFERENCES
528 .PARA
529 .UNDERScore
530 .INDENT -5
531 8.0 LOWER REFERENCES
532 .FILL
533 .PARA
534 .LMARGIN +15
535 .INDENT -15
536 GETPAK.MOD
537 Allocate a small packet.
538 .LMARGIN RESET
539 .EJECT
540 .BTITLE?DFPE1.MOD? BOTTOM ?PAGE ??
541 .TTITLE?DFPE1.MOD?DFGRND.MSI?PAGE ??
542 .PARA
543 .UNDERScore
544 .INDENT -5
545 1.0 MODULE NAME
546 .PARA
547 .FILL
548 DFPE1.MOD
549 .PARA
550 .UNDERScore
551 .INDENT -5
552 2.0 PROGRAMMER
553 .PARA
554 .FILL
555 ETO
556 .PARA
557 .UNDERScore
558 .INDENT -5
559 3.0 DESCRIPTION
560 .PARA

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

561 .FILL
562 .imbed DFPE1.STR G
563 .PARA
564 .UNDERScore
565 .INDENT -5
566 4.0 CALLING SEQUENCE
567 .PARA
568 .FILL
569 .skip
570 .in 10
571 ?SCHED DFPE1,4,...IPACK
572 .skip
573 .PARA
574 .LMARGIN +17
575 .BLOCK BEGIN
576 .INDENT -19
577 IPACK
578 .BREAK
579 Packet pointer to DF request. Packet definition "DF".
580 .skip
581 .BLOCK END
582 .LMARGIN RESET
583 .PARA
584 .UNDERScore
585 .INDENT -5
586 5.0 DISC FILE: USED
587 .PARA
588 .UNDERScore
589 .INDENT -5
590 6.0 EXTERNAL AND COMMON REFERENCES
591 .PARA
592 .UNDERScore
593 .INDENT -5
594 7.0 HIGHER REFERENCES
595 .PARA
596 .UNDERScore
597 .INDENT -5
598 8.0 LOWER REFERENCES
599 .FILL
600 .PARA
601 .LMARGIN +15
602 .INDENT -15
603 GETPAK.MOD
604 Allocate a small packet.
605 .LMARGIN RESET
606 .PARA
607 .LMARGIN +15

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO:IGR.MC68000_DF]DFGRND.FMT_NUM

```

608 .INDENT -15
609 PUTPAK.MOD
610 Release a small packet for reuse.
611 .LMARGIN RESET
612 .EJECT
613 .TITLE?DFPE2.MOD? BOTTOM ?PAGE 1?
614 .TITLE?DFPE2.MOD?DFGRND.MST?PAGE 1?
615 .PARA
616 .UNDERScore
617 .INDENT -5
618 1.0 MODULE NAME
619 .PARA
620 .FILL
621 DFPE2.MOD
622 .PARA
623 .UNDERScore
624 .INDENT -5
625 2.0 PROGRAMMER
626 .PARA
627 .FILL
628 ETO
629 .PARA
630 .UNDERScore
631 .INDENT -5
632 3.0 DESCRIPTION
633 .PARA
634 .FILL
635 .imbed DFPE2.STR/G
636 .PARA
637 .UNDERScore
638 .INDENT -5
639 4.0 CALLING SEQUENCE
640 .PARA
641 .FILL
642 .skip
643 .in 10
644 ?SCHED DFPE2.#4,,,IPACK
645 .SKIP
646 .PARA
647 .LMARGIN +17
648 .BLOCK BEGIN
649 .INDENT -19
650 .IPACK
651 .BREAK
652 Pointer to packet. Packet could either be
653 1) DF rejected message ("DFRJ"), 2) Audio
654 correlation results ("ACC"), or 3) control
  
```

Int.*4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

655 packet with DF results ("CTL" and "DFR").

```

656 .BLOCK END
657 .LMARGIN RESET
658 .PARA
659 .UNDERSCORE
660 .INDENT -5
661 5.0 DISC FILES USED
662 .PARA
663 .UNDERSCORE
664 .INDENT -5
665 6.0 EXTERNAL AND COMMON REFERENCES
666 .PARA
667 .UNDERSCORE
668 .INDENT -5
669 7.0 HIGHER REFERENCES
670 .PARA
671 .UNDERSCORE
672 .INDENT -5
673 8.0 LOWER REFERENCES
674 .FILL
675 .PARA
676 .LMARGIN +15
677 .INDENT -15
678 GETPAK.MOD
679 Allocate a small packet.
680 .LMARGIN RESET
681 .PARA
682 .LMARGIN +15
683 .INDENT -15
684 PUTPAK.MOD
685 Release a small packet for reuse.
686 .LMARGIN RESET
687 .EJECT
688 .BTITLE?FINDOLD.MOD? BOTTOM ?PAGE ??
689 .TTITLE?FINDOLD.MOD?DFGRND.MST?PAGE ??
690 .PARA
691 .UNDERSCORE
692 .INDENT -5
693 1.0 MODULE NAME
694 .PARA
695 .FILL
696 FINDOLD.MOD
697 .PARA
698 .UNDERSCORE
699 .INDENT -5
700 2.0 PROGRAMMER
701 .PARA

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

702 .FILL
703 ETO
704 .PARA
705 .UNDERSORE
706 .INDENT -5
707 3.0 DESCRIPTION
708 .PARA
709 .FILL
710 This module searches the given DF request queue for
711 DF requests that have been in the queue for longer
712 than a set period of time. If an element is found,
713 the element is removed.
714 .PARA
715 .UNDERSORE
716 .INDENT -5
717 4.0 CALLING SEQUENCE
718 .PARA
719 .FILL
720 .skip
721 .in 10
722 JSR FINDOLD
723 .SKIP
724 .PARA
725 .LMARGIN +17
726 .BLOCK BEGIN
727 .INDENT -19
728 An
729 .BREAK
730 Pointer to the header block of the DF request queue.
731 .BLOCK END
732 .LMARGIN RESET
733 .PARA
734 .UNDERSORE
735 .INDENT -5
736 5.0 DISC FILES USED
737 .PARA
738 .UNDERSORE
739 .INDENT -5
740 6.0 EXTERNAL AND COMMON REFERENCES
741 .PARA
742 .FILL
743 .LMARGIN +17
744 .BLOCK BEGIN
745 .INDENT -19
746 HR
747 .BREAK
748 Header block data base for DF request queue.
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO:IGR.MC68000_DF]DFGRND.FMT_NUM

```

749 .BLOCK FWD
750 .LMARGIN RESET
751 .PARA
752 .UNDERScore
753 .INDENT -5
754 7.0 HIGHER REFERENCES
755 .FILL
756 .PARA
757 .LMARGIN +15
758 .INDENT -15
759 DFG2.MOO
760 Remove old DF requests.
761 .LMARGIN RESET
762 .PARA
763 .UNDERScore
764 .INDENT -5
765 8.0 LOWER REFERENCES
766 .FILL
767 .PARA
768 .LMARGIN +15
769 .INDENT -15
770 REMROT.MOO
771 Remove DF request (from bottom).
772 .LMARGIN RESET
773 .EJECT
774 .TITLE?GETCRS.MOO? BOTTOM ?PAGE 1?
775 .TITLE?GETCRS.MOO?DFGRND.MST?PAGE 1?
776 .PARA
777 .UNDERScore
778 .INDENT -5
779 1.0 MODULE NAME
780 .PARA
781 .FILL
782 GETCRS.MOO
783 .PARA
784 .UNDERScore
785 .INDENT -5
786 2.0 PROGRAMMER
787 .PARA
788 .FILL
789 ETO
790 .PARA
791 .UNDERScore
792 .INDENT -5
793 3.0 DESCRIPTI M
794 .PARA
795 .FILL

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO,IGR,MC68000_DF]DFGRND.FMT_NUM

```

796 This module returns the next course DF. The
797 next (top) course DF is "popped" off the coarse
798 DF queue. If the queue is empty this is indicated
799 by the status flag.
800 .PARA
801 .UNDERSCORE
802 .INDENT -5
803 4.0 CALLING SEQUENCE
804 .PARA
805 .FILL
806 .skip
807 .in 10
808 JSR GETCRS
809 .skip
810 .PARA
811 .LMARGIN +17
812 .BLOCK BEGIN
813 .INDENT -19
814 .An
815 .BREAK
816 Pointer to next course DF.
817 .PARA
818 .BLOCK END
819 .BLOCK BEGIN
820 .INDENT -19
821 .Dn
822 .BREAK
823 Status flag. If 0 then no request found.
824 .BLOCK END
825 .LMARGIN RESET
826 .PARA
827 .UNDERSCORE
828 .INDENT -5
829 5.0 DISC FILES USED
830 .PARA
831 .UNDERSCORE
832 .INDENT -5
833 6.0 EXTERNAL AND COMMON REFERENCES
834 .PARA
835 .UNDERSCORE
836 .INDENT -5
837 7.0 HIGHER REFERENCES
838 .FILL
839 .PARA
840 .LMARGIN +15
841 .INDENT -15
842 DFC1 MOD

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

843 DF scheduler.
844 .LMARGIN RESET
845 . PARA
846 . UNDERSCORE
847 . INDENT -5
848 8.0 LOWER REFERENCES
849 . FILL
850 . PARA
851 .LMARGIN +15
852 . INDENT -15
853 GETDF MOD
854 Remove next DF request.
855 .LMARGIN RESET
856 . EJECT
857 .TITLE?GETDF.MOD? BOTTOM PAGE 1?
858 .TITLE?GETDF.MOD?DFGRND.MST?PAGE 1?
859 . PARA
860 . UNDERSCORE
861 . INDENT -5
862 1.0 MODULE NAME
863 . PARA
864 . FILL
865 GETDF MOD
866 . PARA
867 . UNDERSCORE
868 . INDENT -5
869 2.0 PROGRAMMER
870 . PARA
871 . FILL
872 ETO
873 . PARA
874 . UNDERSCORE
875 . INDENT -5
876 3.0 DESCRIPTION
877 . PARA
878 . FILL
879 This module removes (and returns) the next DF request
880 from the given DF request queue. If the queue is empty,
881 this is indicated by the status flag.
882 . PARA
883 . UNDERSCORE
884 . INDENT -5
885 4.0 CALLING SEQUENCE
886 . PARA
887 . FILL
888 skip
889 in 10

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO: IGR: MC68000_DF]DFGRND.FMT NUM

```

890 JSR GETDF
891 SKIP
892 PARA
893 LMARGIN +17
894 BLOCK BEGIN
895 INDENT -19
896 An
897 BREAK
898 Pointer to DF request queue header block.
899 PARA
900 BLOCK END
901 BLOCK BEGIN
902 INDENT -19
903 An
904 BREAK
905 Pointer to DF request.
906 PARA
907 BLOCK END
908 BLOCK BEGIN
909 INDENT -19
910 Dn
911 BREAK
912 Status flag If then queue was empty
913 BLOCK END
914 LMARGIN RESET
915 PARA
916 UNDERSORE
917 INDENT -5
918 5 0 DISC FILE USED
919 PARA
920 UNDERSORE
921 INDENT -5
922 6 0 EXTERNAL AND COMMON REFERENCES
923 PARA
924 FILL
925 LMARGIN +17
926 BLOCK BEGIN
927 INDENT -19
928 HB
929 BREAK
930 Header block data base for DF queues.
931 BLOCK END
932 LMARGIN RESET
933 PARA
934 UNDERSORE
935 INDENT -5
936 7 0 HIGHER REFERENCES

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO: IGR: MC68000 DF]DFGRND.FMT NUM

```

937  FILL
938  PARA
939  LMARGIN +15
940  INDENT -15
941  GETORS MOD
942  Get next coarse DF request
943  LMARGIN RESET
944  PARA
945  LMARGIN +15
946  INDENT -15
947  GETFINE MOD
948  Get next fine DF request
949  LMARGIN RESET
950  PARA
951  UNDERSCORE
952  INDENT -5
953  8 0 LOWER REFERENCES
954  EJECT
955  BTITLE?GETFINE MOD? BOTTOM PAGE 1?
956  BTITLE?GETFINE MOD?DFGRND.MST PAGE 1?
957  PARA
958  UNDERSCORE
959  INDENT -5
960  1 0 MODULE NAME
961  PARA
962  FILL
963  GETFINE MOD
964  PARA
965  UNDERSCORE
966  INDENT -5
967  2 0 PROGRAMMER
968  PARA
969  FILL
970  ETO
971  PARA
972  UNDERSCORE
973  INDENT -5
974  3 0 DESCRIPTION
975  PARA
976  FILL
977  This module searches the fine DF queues in priority
978  order to find the next fine request. Once a page
979  is selected, the next (top) request is "popped" off
980  and returned. If all the fine queues are empty (and
981  therefore no request found), this is indicated by
982  the status flag
983  PARA

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000_DF]DFGRND.FMT_NUM

```

984 .UNDERSORE
985 .INDENT -5
986 4.0 CALLING SEQUENCE
987 .PARA
988 .FILL
989 .skip
990 .in 10
991 JSR GETFINE
992 .SKIP
993 .PARA
994 .LMARGIN +17
995 .BLOCK BEGIN
996 .INDENT -19
997 .An
998 .BREAK
999 Pointer to next line DF request
1000 .PARA
1001 .BLOCK END
1002 .BLOCK BEGIN
1003 .INDENT -19
1004 .Dn
1005 .BREAK
1006 Status flag. If 0 then no request found.
1007 .BLOCK END
1008 .LMARGIN RESET
1009 .PARA
1010 .UNDERSORE
1011 .INDENT -5
1012 5.0 DISC FILES USED
1013 .PARA
1014 .UNDERSORE
1015 .INDENT -5
1016 6.0 EXTERNAL AND COMMON REFERENCES
1017 .PARA
1018 .UNDERSORE
1019 .INDENT -5
1020 7.0 HIGHER REFERENCES
1021 .FILL
1022 .PARA
1023 .LMARGIN +15
1024 .INDENT -15
1025 DFG3 MOD
1026 DF scheduler.
1027 .LMARGIN RESET
1028 .PARA
1029 .UNDERSORE
1030 .INDENT -5

```

In: *4/Output.

In: *2/Output.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

1031 0.0 LOWER REFERENCES
 1032 .FILL
 1033 .PARA
 1034 .LMARGIN +15
 1035 .INDENT -15
 1036 GETDF.MOD
 1037 Remove next DF request.
 1038 .LMARGIN RESET
 1039 .EJECT
 1040 .BTITLE?GETNUM.MID? BOTTOM ?PAGE 1?
 1041 .TTITLE?GETNUM.MID?DFGRND MSI?PAGE 1?
 1042 .PARA
 1043 .UNDERScore
 1044 .INDENT -5
 1045 1.0 MODULE NAME
 1046 .PARA
 1047 .FILL
 1048 GETNUM.MOD
 1049 .PARA
 1050 .UNDERScore
 1051 .INDENT -5
 1052 2.0 PROGRAMMER
 1053 .PARA
 1054 .FILL
 1055 ETO
 1056 .PARA
 1057 .UNDERScore
 1058 .INDENT -5
 1059 3.0 DESCRIPTION
 1060 .PARA
 1061 .FILL
 1062 This module extracts from each DF request queue header
 1063 block the number of DF requests in that queue. These
 1064 numbers are summed to calculate the total number
 1065 of DF requests.
 1066 .PARA
 1067 .UNDERScore
 1068 .INDENT -5
 1069 4.0 CALLING SEQUENCE
 1070 .PARA
 1071 .FILL
 1072 .skip
 1073 .in 10
 1074 JSR GETNUM
 1075 .SKIP
 1076 .PARA
 1077 .LMARGIN +17

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

1078 .BLOCK BEGIN
1079 .INDENT -19
1080 Dm
1081 .BREAK
1082 Total number of DF requests.
1083 .skip
1084 .BLOCK END
1085 .LMARGIN RESET
1086 .PARA
1087 .UNDERScore
1088 .INDENT -5
1089 5.0 DISC FILES USED
1090 .PARA
1091 .UNDERScore
1092 .INDENT -5
1093 6.0 EXTERNAL AND COMMON REFERENCES
1094 .PARA
1095 .FILL
1096 .LMARGIN +17
1097 .BLOCK BEGIN
1098 .INDENT -19
1099 HB
1100 .BREAK
1101 Header block data base for DF queues.
1102 .BLOCK END
1103 .LMARGIN RESET
1104 .PARA
1105 .UNDERScore
1106 .INDENT -5
1107 7.0 HIGHER REFERENCES
1108 .FILL
1109 .PARA
1110 .LMARGIN +15
1111 .INDENT -15
1112 DFG1 MOD
1113 DF request input
1114 .LMARGIN RESET
1115 .PARA
1116 .UNDERScore
1117 .INDENT -5
1118 8.0 LOWER REFERENCES
1119 .EJECT
1120 .BTITLE?GETPTR.MOD? BOTTOM ?PAGE ?
1121 .TTITLE?GETPTR.MOD?DFGRND.MST?PAGE ?
1122 .PARA
1123 .UNDERScore
1124 .INDENT -5
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

1125	1.0	MODULE NAME	
1126	. PARA		
1127	. FILL		
1128	GETPTR.MOD		
1129	. PARA		
1130	. UNDERSCORE		
1131	. INDENT -5		
1132	2.0	PROGRAMMER	
1133	. PARA		
1134	. FILL		
1135	ETO		
1136	. PARA		
1137	. UNDERSCORE		
1138	. INDENT -5		
1139	3.0	DESCRIPTION	
1140	. PARA		
1141	. FILL		
1142		This module returns the pointer to the next available	
1143		element of the given DF queue. The next available element	
1144		is the next open slot to add an element to the queue.	
1145		Since the DF request queues are generally treated as	
1146		stacks, the next available element is the top of the stack.	
1147		If the list is full, the pointer is set to 0.	
1148	. PARA		
1149	. UNDERSCORE		
1150	. INDENT -5		
1151	4.0	CALLING SEQUENCE	
1152	. PARA		
1153	. FILL		
1154	. skip		
1155	. LM 10		
1156	. NOFILL		
1157	MOVE.W	TYPE.Dn	
1158	JSR	GETPTR	
1159	. FILL		
1160	. LM R		
1161	. SKIP		
1162	. PARA		
1163	.LMARGIN +17		
1164	.BLOCK BEGIN		
1165	. INDENT -19		
1166	Dn		
1167	.BREAK		
1168	DF type code.		
1169	. PARA		
1170	.BLOCK END		
1171	.BLOCK BEGIN		

Int*2/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

1172 .INDENT -19
1173   An
1174 .BREAK
1175 Pointer to next available queue element. If 0 then
1176 queue is full.
1177 .BLOCK END
1178 .LMARGIN RESET
1179 .PARA
1180 .UNDERScore
1181 .INDENT -5
1182 5.0 DISC FILE: USED
1183 .PARA
1184 .UNDERScore
1185 .INDENT -5
1186 6.0 EXTERNAL AND COMMON REFERENCES
1187 .PARA
1188 .FILL
1189 .LMARGIN +17
1190 .BLOCK BEGIN
1191 .INDENT -19
1192   HB
1193 .BREAK
1194 Header block data base for DF queues.
1195 .BLOCK END
1196 .LMARGIN RESET
1197 .PARA
1198 .UNDERScore
1199 .INDENT -5
1200 7.0 HIGHER REFERENCES
1201 .FILL
1202 .PARA
1203 .LMARGIN +15
1204 .INDENT -15
1205 DFGL MOD
1206 DF request input
1207 .LMARGIN RESET
1208 .PARA
1209 .UNDERScore
1210 .INDENT -5
1211 8.0 LOWER REFERENCES
1212 .EJECT
1213 .BTITLE?PUSH,MOD? BOTTOM ?PAGE ??
1214 .TTITLE?PUSH,MOD?DFGRND,MT?PAGE,??
1215 .PARA
1216 .UNDERScore
1217 .INDENT -5
1218 1.0 MODULE NAME
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000_DF]DFGRND.FMT_NUM

```

1219 .PARA
1220 .FILL
1221 .PUSH MOD
1222 .PARA
1223 .UNDERSCORE
1224 .INDENT -5
1225 2.0 PROGRAMMER
1226 .PARA
1227 .FILL
1228 ETO
1229 .PARA
1230 .UNDERSCORE
1231 .INDENT -5
1232 3.0 DESCRIPTION
1233 .PARA
1234 .FILL
1235 This module adds a DF request to the given DF request
1236 queue. Since the DF request queues are generally implemented
1237 as stacks, the given request is "pushed" onto the top of
1238 the stack. When the entry is added to the queue, it is tagged
1239 with the current "time" from a variable in local.
1240 .PARA
1241 .UNDERSCORE
1242 .INDENT -5
1243 4.0 CALLING SEQUENCE
1244 .PARA
1245 .FILL
1246 .skip
1247 .nofill
1248 .in 10
1249 JSR PUSH
1250 .fill
1251 .skip
1252 .PARA
1253 .LMARGIN +17
1254 .BLOCK BEGIN
1255 .INDENT -19
1256 An
1257 .BREAK
1258 Pointer to packet containing DF request. Packet definition
1259 "DF" or "DC".
1260 .PARA
1261 .BLOCK END
1262 .BLOCK BEGIN
1263 .INDENT -19
1264 An
1265 .BREAK

```

Int.*4/Input

Int.*4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFGRND.FMT_NUM

```

1266 Pointer to next available element on DF queue.
1267 .PARA
1268 .BLOCK END
1269 .BLOCK BEGIN
1270 .INDENT -19
1271 .An
1272 .BREAK
1273 Pointer to header block of DF request.
1274 .BLOCK END
1275 .LMARGIN RESET
1276 .PARA
1277 .UNDERScore
1278 .INDENT -5
1279 5.0 DISC FILES USED
1280 .PARA
1281 .UNDERScore
1282 .INDENT -5
1283 6.0 EXTERNAL AND COMMON REFERENCES
1284 .PARA
1285 .FILL
1286 .LMARGIN +17
1287 .BLOCK BEGIN
1288 .INDENT -19
1289 HB
1290 .BREAK
1291 Header block data base for DF queues.
1292 .BLOCK END
1293 .LMARGIN RESET
1294 .PARA
1295 .UNDERScore
1296 .INDENT -5
1297 7.0 HIGHER REFERENCES
1298 .FILL
1299 .PARA
1300 .LMARGIN +15
1301 .INDENT -15
1302 DFG1.MOD
1303 DF request input
1304 .LMARGIN RESET
1305 .PARA
1306 .UNDERScore
1307 .INDENT -5
1308 8.0 LOWER REFERENCES
1309 .EJECT
1310 .BTITLE?REMBOT.MID? BOTTOM ?PAGE ??
1311 .TITLE?REMBOT.MID?DFGRND.MST?PAGE ??
1312 .PARA

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000 DF]DFGRND.FMT_NUM

```

1313 .UNDERSORE
1314 .INDENT -5
1315 1.0 MODULE NAME
1316 .PARA
1317 .FILL
1318 REMBOT MOD
1319 .PARA
1320 .UNDERSORE
1321 .INDENT -5
1322 2.0 PROGRAMMER
1323 .PARA
1324 .FILL
1325 ETO
1326 .PARA
1327 .UNDERSORE
1328 .INDENT -5
1329 3.0 DESCRIPTION
1330 .PARA
1331 .FILL
1332 This module removes a DF request from the bottom of
1333 the given DF request queue. This module updates the
1334 pointers and counter in the given DF request queue
1335 header block to delete the element. If the DF request
1336 that was removed originated from the Perkin Elmer, a
1337 rejected message is formatted and DFPE2 is scheduled.
1338 .PARA
1339 .UNDERSORE
1340 .INDENT -5
1341 4.0 CALLING SEQUENCE
1342 .PARA
1343 .FILL
1344 .skip
1345 .lm 10
1346 .nofill
1347 JSR REMBOT
1348 .fill
1349 .skip
1350 .lm r
1351 .PARA
1352 .LMARGIN +17
1353 .BLOCK BEGIN
1354 .INDENT -19
1355 An
1356 .BREAK
1357 Pointer to header block of DF queue.
1358 .skip
1359 .BLOCK END
  
```

Int.4/Input

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFGRND.FMT NUM

```

1360 .LMARGIN RESET
1361 .PARA
1362 .UNDERSCORE
1363 .INDENT -5
1364 5.0 DISC FILE USED
1365 .PARA
1366 .UNDERSCORE
1367 .INDENT -5
1368 6.0 EXTERNAL AND COMMON REFERENCES
1369 .PARA
1370 .FILL
1371 .LMARGIN +17
1372 .BLOCK BEGIN
1373 .INDENT -19
1374 HB
1375 .BREAK
1376 Header block data base for DF queues.
1377 .BLOCK END
1378 .LMARGIN RESET
1379 .PARA
1380 .UNDERSCORE
1381 .INDENT -5
1382 7.0 HIGHER REFERENCES
1383 .FILL
1384 .PARA
1385 .LMARGIN +15
1386 .INDENT -15
1387 DFG1.MOD
1388 DF request input
1389 .LMARGIN RESET
1390 .PARA
1391 .LMARGIN +15
1392 .INDENT -15
1393 FINDOLD.MOD
1394 Find and remove old DF requests
1395 .LMARGIN RESET
1396 .PARA
1397 .UNDERSCORE
1398 .INDENT -5
1399 8.0 LOWER REFERENCES
1400 .FILL
1401 .PARA
1402 .LMARGIN +15
1403 .INDENT -15
1404 GETPAK.MOD
1405 Allocate a small packet.
1406 .LMARGIN RESET

```

Database

Improved GUARDRAIL V MC68000 'DF' Files
DRCL: [ALGO. IGR. MC68000_DF]DFGRND.FMT_NUM

1407 .EJECT

DFIN.FMT_NUM

16.18

***** Source Listing ==> DFIN.FMT_NUM *****

1 LLEN 116

2 MODULE NAME

3

4 DFIN.MOD

5

6 PROGRAMMER

7

8 P. Wong

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

DESCRIPTION

This routine receives messages from another CPU. It is executed when the airborne DF CPU is interrupted. The first byte of the global buffer used to pass the message holds the packet size (large or small) of the message. The correct size local packet is allocated and the message is moved into it. The action routine for the incoming packet is retrieved from the TIB table using the offset passed in the global buffer. That routine is then scheduled with the packet.

CALLING SEQUENCE

Initiated by interrupt from another CPU.

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

ACPUTIB Database
 Table of TIB pointers for all routines on the airborne CPUs.

BUPTIBSY Flag
 Buffer in use indicator.

BUPTIF Buffer
 Global buffer to receive messages.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFIN.FMT NUM

44 *
 45 *
 46 *
 47 *
 48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *
 59 *
 60 *
 61 *
 62 *

BUF:FDON

Flag

Indicates buffer is completely full. This allows detection of spurious interrupts if it is not set.

BUF:DOFF

Integer

Offset into the TIR pointer table for the desired routine on the ADF CPU.

HIGHER REFERENCES

LOWERREFERENCES

ADF:IT.MOD

DF CPU Bite Job

PAC:BUF.MOD

Move data from inter-CPU buffer into allocated packet.

16.19 DFOUT.FMT_NUM

***** Source Listing --> DFOUT.FMT_NUM *****

1 LLEN 116
 2 MODULE NAME

3
 4 DFOUT.MOD

5
 6 PROGRAMMER

7
 8 WORK

DESCRIPTION

This routine sends message packets to the airborne DF CPU. This routine will be initiated by the scheduling of any routine on the ADF CPU from another CPU. The message is written into a global buffer so the local packet can be released. The ADF CPU is then interrupted so it can read the waiting message.

CALLING SEQUENCE

?SCHED ADF CPU routine, #?..., PACADDR

PACADDR INT*4/INPUT
 Pointer to any message packet

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

BUFBSY Flag
 Buffer in use indicator.

BUFWHO Flag
 Identifies which CPU is currently using BUFDF.

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000 DF] DFOUT.FMT NUM

44 *
 45 *
 46 *
 47 *
 48 *
 49 *
 50 *
 51 *
 52 *
 53 *
 54 *
 55 *
 56 *
 57 *
 58 *
 59 *
 60 *
 61 *
 62 *
 63 *
 64 *
 65 *
 66 *
 67 *
 68 *
 69 *

BUF:OP

Buffer
 Global buffer to pass messages to the DF
 CPU.

BUF:BDON

Flag
 Indicates if buffer is completely full.
 This allows detection of spurious
 interrupts.

BUF:OFF

Integer
 Offset into the TIB pointer table for the
 desired routine on the ADF CPU

ACT:TIB

Database
 Table of TIB pointers for all routines the
 airborne CPUs

HIGHER REFERENCES

LOWERREFERENCES

RELPAK.MOD
 BUFPAK.MOD

Release any valid combination of packets.
 Move data from a local packet to a global
 inter-CPU buffer

Improved GUARDRAIL V MC68000 'DF' Files
 DFC1:|ALGO.IGR.MC68000_DF|DFPE1.FMT NUM

DFPE1.FMT_NUM

16 20

***** Source Listing -> DFPE1.FMT_NUM *****

 LLEN 116
 MODULE NAME

DFPE1.MOD

PROGRAMMER

P. Eto

DESCRIPTION

DFPE1

This job is scheduled by the Perkin Elmer message input handler (see GPIN) when a DF request is received. This job formats the request message and then schedules the DF Request Input job (see DFC1).

CALLING SEQUENCE

?SCHED DFPE1,#4,...,IPACK

IPACK

Int*4/Input
 Packet pointer to DF request. Packet
 definition "DP".

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO,IGR,MC68000_DF]DFPEL.FMT NUM

44 * LOWERREFERENCES

45 * GETPAK.MOD Allocate a small packet.

46 * DFCI.MOD DF request input

47 * PUTPAK.MOD Release a small packet for reuse.

DFPE2.FMT_NUM

16 21

***** Source Listing --> DFPE2.FMT_NUM *****

LLFN 116

MODULE NAME

DFPE2.MOD

PROGRAMMER

M. Swanson

DESCRIPTION

DFPE2

This job is either scheduled by the Remove Old DF requests job (DFG2) with a DF rejected message, the Downlink Handler (GAIN) with DF results or a timeout message, or the Audio Correlator Input Job (DFG9) with the Audio Correlation results. If this job was scheduled by the Remove Old DF requests job, it will simply format a DF rejected message to pass to the Perkin Elmer Output job (GPOUT). If this job was scheduled by the Downlink Handler (GAIN), then the message either contains two DF responses, a DF response and a timeout, or two timeout messages. In either case, the results are formatted into a message to be sent to the Perkin Elmer Output job (GPOUT). If the DF request did not require audio correlation, the message is sent. If the DF request required audio correlation, this job will wait for the audio correlation results before sending the message. In the case of an Interop, DF when a response is to be sent to the Perkin Elmer (either the DF completed/rejected or was rejected), the DF scheduler (DFIT) is scheduled.

CALLING SEQUENCE

?SCHED DFPE2,#4,...,IPACK

IPACK Int.*4/Input
 Pointer to packet. Packet could either be

Improved GUARDRAIL V MC68000 'DF' Files DRCL: [ALGO.IGR.MC68000_DF]DFPE2.FMT NUM

1) DF rejected message ("DFRJ"), 2) Audio correlation results ("ACT"), or 3) control packet with DF results ("CTL" and "DFR").

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

INT-ROP Flag
 Interop DF flag received.

HIGHER REFERENCES

GATIM.MOD Response queue timeout handler.
 DFGI.MOD Remove old DF requests.
 DFGI.MOD Audio correlator control.
 REMPT.MOD Remove DF request (from bottom).

LOWER REFERENCES

GETIPAK.MOD Allocate a large packet.
 GETPAK.MOD Allocate a small packet.
 GPOUT.MOD Queue messages to be sent to the Perkin Elmer.
 RELPAK.MOD Release any valid combination of packets.
 DFGI.MOD DF scheduler.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: |ALGO.IGR.MC68000_DF|DFPON.FMT NUM

DFPON.FMT_NUM

16 22

***** Source Listing --> DFPON.FMT_NUM *****

1 LLF# 116

2 MODULE NAME

4 DFPON.MOD

7 PROGRAM R

9 ETC

DESCRIPTION

This module performs the power on initialization for the DF data collection. The SIO and FIO are set up for use with the Fast DF Controller. A DRCH is set up so that the DF interrupt service routine can perform ARTE calls. The Fast DF Controller is also initialized.

CALLING SEQUENCE

JSR DFPON

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

LOWERREFERENCES

INITDF.MOD Initialize the DF data collection system.

ACUNIT.MOD Initialize the FIO for use with ACUs.

Improved GUARDRAIL V MC68000 'DF' Files

17 FILES WITH EXTENSION '.LST'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000 DF]DFA1.LST

17.1 DFA1.LST

PAGE 1

ERR LINE ADDR DATE= 08/13/85 TIME= 10:47:33 DFA1

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

TTL 'DFA1'

VERSION DATE
 1.0 6/16/83
 1.1 5/23/84
 1.2 8/20/84 CHECK FOR REPORT ENABLE

*INCLUDE DFA1.FMT/G
 LLEN 116
 MODULE NAME
 DFA1.MOD

PROGRAMMER
 ETO

DESCRIPTION

DFA1

The DF Request Input job receives DF Request messages from the Uplink Message Handler (AGIN). There are two states that a DF request will assume while in the ADPU. The "new" DF is the DF request that has just been received from the GDPU. The "current" DF is the DF request that is being processed. The DF request parameters will be made available as the new DF to be performed. DF segments will be synchronized by the receipt of new DF requests from the GDPU. This job starts the process on the next DF by scheduling the Start/Continue DF job (DFA2). The "done with current" flag indicates whether or not the last (current) DF completed prior to the receipt of the new request. During normal operation, the current DF should complete before the new DF request is received from the GDPU. If a DF request is received before the current DF has completed, an error message is generated to indicate the problem. The Nav. Data Input job (DFA4) is

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000.DF]DFAL.LST

```

44 *
45 *      scheduled after a timeout to read the current navigation data
46 *      during the middle of the DF data collection.
47 *
48 *
49 *      CALLING SEQUENCE
50 *
51 *      ?SCHED DFAL.#2,...,IPACK
52 *
53 *      IPACK      Int*4/Input
54 *                  Pointer to packet containing DF request.
55 *                  Packet definition "DF".
56 *
57 *
58 *      DISC FILES USED
59 *
60 *
61 *      EXTERNAL AND COMMON REFERENCES
62 *
63 *
64 *      HIGHER REFERENCES
65 *
66 *
67 *      LOWERREFERENCES
68 *
69 *
70 *      I/O DEVICES USED: NONE
71 *
72 *      ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
73 *      ADMINIC
74 *
75 *      METHOD:
76 *      PDL:
77 *
78 *      ?DLINK IPACK      /* GET DF REQUEST */
79 *
80 *      /* IPACK POINTS TO A PACKET CONTAINING DF REQUEST */
81 *
82 *      ?SCHED DFA4.#3,#1,#1 /* TIMEOUT FOR NAV DATA INPUT */
83 *      IF ARFID = ARF1
84 *      THEN /* CHECK FLAG FOR ARF1 */
85 *      IF ARF1 ENABLED /* CHECK FLAG IN DF MESSAGE */
86 *      THEN
87 *      DFRPEN <- TRUE /* SET FLAG */
88 *      ELSE
89 *      DFRPEN <- FALSE /* ELSE DISABLE */
90 *      ENDIF

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFA1.LST

```

91 * ELSE /* THIS IS ARF2 */
92 * IF ARF2 ENABLED
93 * THEN
94 *   DFRPEN <- TRUE
95 * ELSE
96 *   DFRPEN <- FALSE
97 * ENDIF
98 *
99 * IF TUNE AUDIO CORR RCVR /* CHECK FLAG IN REQUEST */
100 * THEN /* TUNE IT */
101 *   FORMAT TUNING STRING
102 *   ?WRITM D:OB,#N,BUFAD /* SEND STRING TO RCVR */
103 *   SET TO A:ATTACK
104 *   ?WRITM D:OB,#N,BUFAD /* TUNE THE RCVR */
105 * ENDIF
106 *
107 * ?SCHED DFA2,#2,,IPACK /* SCHED DFA2, START DATA COLL */
108 *
109 * ?EXIT
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *
190 *
191 *
192 *
193 *
194 *
195 *
196 *
197 *
198 *
199 *
200 *
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *
265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFAL.LST

```

265 * significant digit represents 1 GHZ.
266 *
267 * 2) DF_GAIN RECEIVER GAIN SETTINGS
268 * This field is ignored by ADPU software.
269 *
270 * 3) DF_BAND BANDWIDTH VALUES
271 * 0 = 8 KHZ
272 * 1 = 15 KHZ
273 * 2 = 50 KHZ
274 *
275 * 4) DF_DET DETECTOR VALUES
276 * 0 = FM
277 * 2 = AM
278 * 4 = CW
279 * 5 = LSB
280 * 6 = USB
281 * 7 = ISB
282 *
283 * 5) DF_TC AGC TIME CONSTANT
284 * This field is ignored by ADPU software
285 *
286 * 6) DF_DU AGC DUMP CONSTANT
287 * This field is ignored by ADPU software.
288 *
289 * 7) DF_SQ SQUELCH
290 * This field is ignored by ADPU software.
291 *
292 * 8) DF_CV COARSE VERIFY FLAG
293 * This flag is used only during calibration verify.
294 * 0 = fine DF
295 * 1 = coarse DF
296 *
297 * 9) DF_POL ANTENNA POLARIZATION
298 * This field is ignored by ADPU software.
299 *
300 * 10) DF_TYP DF TYPE DESIGNATOR
301 * 1 = BITE
302 * 2 = MANUAL DF
303 * 3 = INTEROP
304 * 4 = AUTO DF (DS)
305 * 5 = GENERAL SEARCH (FINE)
306 * 6 = CALIBRATION
307 * 7 = AOI
308 * 8 = GEOSCREEN
309 *
310 * 11) DF_AUDIO AUDIO CORRELATOR RECEIVER TUNING
311 * 0 = do not tune audio correlator receiver

```

Improved GUARDRAIL V MC68000 'DF' Files BRC1:[ALGO.IGR.MC68000.DF]DFA1.LST

```

312 * 1 = tune audio correlator receiver
313 *
314 * 1.3) DF_FINE (COARSE/FINE DF FLAG)
315 * 0 = coarse DF
316 * 1 = fine DF
317 *
318 * 1.4) DF_RESP RESPONSE FLAG
319 * 0 = no response
320 * 1 = send response
321 *
322 * 1.4) DF_CONT CONTIGUOUS FLAG
323 * 0 = non contiguous DF
324 * 1 = contiguous segments---ie. do no re-tune receiver
325 *
326 * 1.5) DF_SEG DF SEGMENT NUMBER
327 * Valid segment numbers are from 1 thru 6.
328 *
329 * 1.6) DF_EN1_DF_EN2 ENABLE ARFX REPORTING
330 * 0 = Do not send additional DF data messages (UDR1,UDR2,UDR3)
331 * 1 = Do send additional DF data messages.
332 *
333 DF_MID EQU 0 MESSAGE ID ($12)
334 DF_NIB EQU 1 ARF ID. & BLOCK COUNT
335 DF_ARF EQU 3<<8+4 ARF ID.
336 DF_PLK EQU 3<<8+0 BLOCK COUNT
337 DF_ACC EQU 2 ACCOUNTABILITY
338 DF_FRQ EQU 4 1ST 2 BCD DIGITS
339 DF_FR1 EQU 4 2ND 2 BCD DIGITS
340 DF_FR2 EQU 5 3RD 2 BCD DIGITS
341 DF_FR3 EQU 6 4TH 2 BCD DIGITS
342 DF_FR4 EQU 7 LAST 2 BCD DIGITS
343 DF_FR5 EQU 8 GAIN TYPE & MANUAL CNTRL
344 DF_GAIN EQU 9 GAIN TYPE
345 DF_TYPE EQU 1<<8+6 MANUAL GAIN SETTING
346 DF_MAN EQU 5<<8+0 BANDWIDTH AND DETECTORS
347 DF_RCVR EQU 10 BANDWIDTH CODE SEE NOTE
348 DF_RNG EQU 10 DETECTOR CODE SEE NOTE 3
349 DF_RAND EQU 2<<8+5 AGC TIME CONSTANT
350 DF_DET EQU 2<<8+2 AGC DUMP CONSTANT
351 DF_TC EQU 0<<8+1 MISC FLAGS
352 DF_DU EQU 0<<8+0 SQUELCH
353 DF_FLAGS EQU 11 ENABLE ARF1 REPORTING
354 DF_Q EQU 0<<8+7 ENABLE ARF2 REPORTING
355 DF_FN1 EQU 0<<8+5 COARSE VERIFY FLAG
356 DF_FN2 EQU 0<<8+4 ANTENNA POLARIZATION
357 DF_FV EQU 0<<8+3
358 DF_FNL EQU 1<<8+1

```

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000 DF]DFAIL.LST

```

359 0000000C
360 00000000
361 00000007
362 00000006
363 00000005
364 00000004
365 00000300
366 0000000E
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405

```

```

DF_TYP EQU 12 DF_TYPE SEE NOTE 4
DF_ASUB EQU 13 DF_SUP TYPE & AUDIO CORR
DF_AUDIO EQU 0<<8+7 AUDIO CORR RCVR TUNE
DF_FINE EQU 0<<8+6 COARSE/FINE FLAG
DF_RESP EQU 0<<8+5 RESPOND FLAG
DF_CONT EQU 0<<8+4 CONTIGUOUS FLAG
DF_REG EQU 3<<8+0 SEGMENT NUMBER
DF EQU 14
*
*SINCLUDE DFR.PAC/G
*
* DFR DF RESPONSE (ADPU -> GDFU)
*
* MESSAGE TO TRANSFER DF RESPONSE INFORMATION
*
* 1) DFR_ERR DF ERRORS BIT MAP BIT ASSIGNMENTS
* 0 - DF REJECTED (PREV. TOO LONG)
* 1 - DF DID NOT COMPLETE IN SEGMENT
* 2 - NO SIGNAL PRESENT
* 3 - BASE LINE SIG PRES FAILED
* 4 - BASE LINE SIG PRES ABORT
* 5 - ROLL ANGLE EXCEEDS LIMIT
* 6 - MULTIPLE MINIMA IN 1ST SEARCH
* 7 - POOR FIT, BAD QUALITY
* 8 - MISSING SEGMENT (FINE DF)
* 9 - NO CAL TABLES
*
* 2) DFR_TYP DF TYPE VALUES
*
* 1 - BITE
* 2 - MANUAL DF
* 3 - INTEROP
* 4 - AUTO DF (DS)
* 5 - GENERAL SEARCH (FINE)
* 6 - CALIBRATION
* 7 - AOI
* 8 - GEOSCREEN
*
* 3) DFR_SSM SIGNAL STRENGTH MEASURE
* This field contains the digitized AGC voltage which is a
* crude measure of signal strength. Division of this field
* by 1.5 should give and approximation of SNR in dB. This
* field is only valid for calibration measurements.
*
* 4) DFR_LAT LATITUDE
* Latitude of ARF at time of measurement in BAM.
*

```


Improved GUARDRAIL V MC68000 'DF' Files BRC1:[ALGO.IGR.MC68000 DF]DFA1.LST

```

406 * 1) DFR_LON LONGITUDE
407 * Longitude of ARF at time of measurement in BAM.
408 *
409 * 6) DFR_LOB LINE OF BEARING
410 * Line of bearing to emitter with respect to ARF heading.
411 *
412 * 7) DFR_QUAL QUALITY OF CAL TABLE FIT
413 * Quality of the base line measurements cal table fit. Valid
414 * values are 0 thru 32 with 0 being a perfect fit to the cal
415 * table.
416 *
417 * 8) DFR_HEA HEADING
418 * Heading of ARF at time of measurement
419 *
420 * 9) DFR_ROLL ROLL ANGLE
421 * Roll angle of ARF at time of measurement
422 *
423 * 10) DFR_BASE BASELINE PHASE MEASUREMENTS
424 * Baseline phase measurements: 5 values for VHF LO, 8
425 * values for VHF HI and 6 values for UHF.
426 *
427 DFR_MID EQU 0 MESSAGE ID ($15)
428 DFR_NIB EQU 1 ARF ID & BLOCK COUNT
429 DFR_ARF EQU 3<<8+4 ARF ID
430 DFR_BLK EQU 3<<8+0 BLOCK COUNT
431 DFR_ACC EQU 2 ACCOUNTABILITY
432 DFR_ERR EQU 4 ERROR CODE SEE NOTE 1
433 DFR_TYP EQU 6 DF TYPE SEE NOTE 2
434 DFR_SSM EQU 7 SIGNAL STRENGTH MEASURE
435 DFR_LAT EQU 8 LATITUDE (BAM)
436 DFR_LON EQU 12 LONGITUDE (BAM)
437 DFR_LOB EQU 16 LINE OF BEARING (BAM)
438 DFR_QUAL EQU 18 QUALITY
439 DFR_HEA EQU 20 HEADING (BAM)
440 DFR_ROLL EQU 22 AIRCRAFT ROLL (BAM)
441 DFR_BASE EQU 24 5, 6 OR 8 BASELINE PHASE
442 DFR EQU 40
443 *
444 *SINCLUE ARTEMACS.S/G NOLIST,?DLINK,?EXIT,?SCHED,?WRITH
445 *SPART NOLIST
446 * LIST
447 *
448 *SINCLUE ARTEDATA.S/G NOLIST,RCB
449 *SPART NOLIST
450 * LIST
451 *
452 * LIST
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```

782	00000000
783	00000001
784	00000002
785	00000003
786	00000004
787	00000005
788	00000006
789	00000007
790	00000008
791	00000009
792	00000010
793	00000011
794	00000012
795	00000013
796	00000014
797	00000015
798	00000016
799	00000017
800	00000018
801	00000019
802	00000020
803	00000021
804	00000022
805	00000023
806	00000024
807	00000025
808	00000026
809	00000027
810	00000028
811	00000029
812	00000030
813	00000031
814	00000032
815	00000033
816	00000034
817	00000035
818	00000036
819	00000037
820	00000038
821	00000039
822	00000040
823	00000041
824	00000042
825	00000043
826	00000044
827	00000045
828	00000046
829	00000047
830	00000048
831	00000049
832	00000050
833	00000051
834	00000052
835	00000053
836	00000054
837	00000055
838	00000056
839	00000057
840	00000058
841	00000059
842	00000060
843	00000061
844	00000062
845	00000063
846	00000064
847	00000065
848	00000066
849	00000067
850	00000068
851	00000069
852	00000070
853	00000071
854	00000072
855	00000073
856	00000074
857	00000075
858	00000076
859	00000077
860	00000078
861	00000079
862	00000080
863	00000081
864	00000082
865	00000083
866	00000084
867	00000085
868	00000086
869	00000087
870	00000088
871	00000089
872	00000090
873	00000091
874	00000092
875	00000093
876	00000094
877	00000095
878	00000096
879	00000097
880	00000098
881	00000099
882	00000100
883	00000101
884	00000102
885	00000103
886	00000104
887	00000105
888	00000106
889	00000107
890	00000108
891	00000109
892	00000110
893	00000111
894	00000112
895	00000113
896	00000114
897	00000115
898	00000116
899	00000117
900	00000118
901	00000119
902	00000120
903	00000121
904	00000122
905	00000123
906	00000124
907	00000125
908	00000126
909	00000127
910	00000128
911	00000129
912	00000130
913	00000131
914	00000132
915	00000133
916	00000134
917	00000135
918	00000136
919	00000137
920	00000138
921	00000139
922	00000140
923	00000141
924	00000142
925	00000143
926	00000144
927	00000145
928	00000146
929	00000147
930	00000148
931	00000149
932	00000150
933	00000151
934	00000152
935	00000153
936	00000154
937	00000155
938	00000156
939	00000157
940	00000158
941	00000159
942	00000160
943	00000161
944	00000162
945	0000016

*\$INCLUDE DFERRS.EQU

- * * *
- * * * DF ERRORS BIT ASSIGNMENTS

0	DF REJECTED (PREV. TOO LONG)
1	DF DID NOT COMPLETE IN SEGMENT
2	NO SIGNAL PRESENT
3	BASE LINE SIG PRES FAILED
4	BASE LINE SIG PRES ABORT
5	ROLL ANGLE EXCEEDS LIMIT
6	MULTIPLE MINIMA IN 1ST SEARCH
7	POOR FIT, BAD QUALITY
8	MISSING SEGMENT (FINE DF)
9	NO CAL TABLES

SECT **AADMINUC**

PACHEAD OFFSET
BLOCK COUNT

POINTER TO PARAM. BLOCK

[illegible]

1ST 4 BCD DIGITS
2ND 4 BCD DIGITS
AGC SELECT
BAND WIDTH, DET., A
GHZ BIT

1ST WORD OF FREQ
2ND WORD OF FREQ.
AGC / MGC
DET., BW, AGC TC, AGC DUMP
GHZ BIT

AGC TIME CONSTANT BIT
AGC DUMP/ATTACK BIT
GHZ FIELD MASK

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFAL.LST

829	000000001	ENABLE	EQU	1	ENABLE FLAG
830	000000001	ARF1	EQU	1	ARF1 ID VALUE
831		*			
832	000000028	DFALST	?DLINK	IPACK	: GET PACKET POINTER
844	000000038		?SCHED	DFAA,#5,#1,#1	
879	000000050		MOVE.L	IPACK,A2	GET PACKET POINTER
880	000000054		CLR.L	D2	CLEAR OUT INITIAL BIT VALUE
881	000000056		MOVE.W	#DF EN1,D2	ASSUME ARF1
882	00000005A		MOVE.B	DF_FLAGS(A2),D0	GET OVERALL FIELD
883	00000005E		CMPI.B	#ARF1,ARFID	CHECK IF ARF1
884	000000066		REQ.S	CHECK	IT IS SO GO CHECK
885	000000068		MOVE.W	#DF EN2,D2	SET BIT FOR ARF2
886	00000006C	CHECK	BTST	D2,D0	CHECK IF REPORTING ENABLED
887	00000006E		REQ.S	DISARF	DISABLE THE REPORTING
888	000000070		MOVE.W	#ENABLE,DFRPN	ENABLE REPORTING
889	000000078		BRA.S	CHKAC	
890	00000007A		CLR.W	DFRPN	DISABLE REPORTING
891	000000080	DISARF	MOVE.B	DF_ASUB(A2),D0	GET SUBTYPE
892	000000084	CHKAC	BTST	#DF AUDIO,D0	CHECK IF ACCOR TUNE
893	000000088		BEQ	DONE	DO NOT TUNE, FINISHED
894	00000008C		LEA	DF_FRI(A2),A0	GET POINTER TO 1ST BYTE OF FREQ
895	000000090		LEA	TSTRING,A1	GET POINTER TO OUTPUT BUFFER
896	000000094		JSR	FMTFRQ	FORMAT FREQUENCY
897	00000009A		MOVE.B	DF_RNG(A2),D0	GET BAND WIDTH AND DETECTOR
898	00000009E		BSET	#AGCTC,D0	MAKE SURE SLOW TIME CONST.
899	0000000A2		BCLR	#ACCDUMP,D0	REQUEST AGC DUMP
900	0000000A6		MOVE.W	D0,RNG	PUT INTO TUNING COMMAND
901	0000000AC	0016 R	MOVE.B	DF_FRI(A2),D1	GET MSB OF FREQ.
902	0000000B0		CLR.W	GHZ	CLEAR GHZ FLAG
903	0000000B6	0018 R	AND.W	#GHZFID,D1	MASK OFF GHZ FIELD
904	0000000BA		REQ.S	NOGHZ	NO GHZ FIELD
905	0000000BC		MOVE.W	#1,GHZ	ELSE, SET GHZ FLAG
906	0000000C4	NOGHZ	?WRITW	DSOB,#N,BUFAD,..D0	
946	0000000F2		BSET	#ACCDUMP,D0	REQUEST AGC ATTACK
947	0000000F6		MOVE.W	D0,RNG	PUT BACK INTO TUNING COMMAND
948					
949					
950					
951					
952	0000000FC		MOVE.W	#COUNT,D1	GET LOOP COUNT
953	000000100		DBF	D1,WAIT	SPIN LOOP
954					
955	000000104		?WRITW	DSOB,#N,BUFAD	
995					

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000 DF]DFA1.LST

996 0000012A 2SCHED DFA2.2.20.21.1PACK : SCHEDULE DFA2

1031
 1032 00000144
 1046
 1047 0000014E

DONE
 *
 *
 ?EXIT
 ?EXIT
 END

ASSEMBLER ERRORS = 0

SYMBOL TABLE

ACCMID	00000000	ACCMID	00000001	ACCDUMP	00000000
ACCTC	00000001	ACSEND	E 00000001	AINTMID	00000002
ALOBMID	00000003	ARF1	00000001	ARFID	E 00000008
ANFTMID	00000004	ANTAMID	00000005	ASTRMID	00000006
BANGMID	00000007	BOFMID	0000000E	BITEMID	00000009
BITMID	00000008	BITQID	0000000A	BLSCA	00000004
BLSGF	00000003	BTRMID	0000000B	BTRQID	0000000C
BUFAC	R 00000004	CALCMID	0000000D	CHECK	R 0000000C
CHKAC	R 00000080	COUNT	0000001D	CPERMID	0000000F
CPURMID	00000010	CTLMID	00000011	DF	0000000E
DFA1ST	R 00000028	DFA2	E 00000002	DFA4	E 00000000
DFPMID	00000013	DEMID	00000012	DFPMID	00000014
DFR	00000028	DEREJ	00000000	DFRMID	00000016
DFRMID	00000015	DERMIB	00000004	DFRPN	E 00000007
DFRACC	00000002	DFRARF	00000030	DFRBASE	00000018
DFRBLK	00000300	DFRERR	00000004	DFRHEA	00000014
DFRLAT	00000008	DFRLOB	00000010	DFRLON	0000000C
DFRMID	00000000	DFRNTB	00000001	DFRQUAL	00000012
DFRROLL	00000018	DFRSSH	00000007	DFRTYP	00000006
DFACC	00000002	DFARF	00000030	DFASUR	0000000D
DFAUDIO	00000007	DFBAND	00000205	DFBLK	00000300
DFCONT	00000004	DFCV	00000003	DFDET	00000202
DFDU	00000000	DFEN1	00000005	DFEN2	00000004
DFFINE	00000006	DFFLAGS	00000008	DFFR1	00000004
DFFR2	00000005	DFFR3	00000006	DFFR4	00000007
DFFR5	00000008	DFFRQ	00000004	DFGAIN	00000009
DFMAN	00000500	DFMID	00000000	DFNIB	00000001
DFPOL	00000101	DFRCVR	0000000A	DFRESP	00000005
DFRNG	0000000A	DFSEG	00000300	DFSQ	00000007
DFTC	00000001	DFTYP	0000000C	DFTYPE	00000106
DFMID	00000017	DISARF	R 0000007A	DONE	R 0000012A
DFMID	00000018	DISARMID	00000019	DSACMID	0000001A
DSDOMID	0000001B	DSRMID	0000001C	DSOB	E 00000003
DSPMID	0000001D	DSPOMID	0000001E	DSPRMID	0000001F
DSRMID	00000020	ENABLE	00000001	FMIFRO	E 00000006
FMILT	R 0000001C	GTPAK	E 00000004	GHZ	R 00000018

Improved GUARDRAIL V MC68000 'DF' Files
 BRCL:[ALGO,IGR,MC68000_DF]DFAL.LST

GRZFLD	000000F0	GI*TMID	00000021	GLOBMID	00000022
GPROJ MID	00000023	G*VCMID	00000025	GSACMID	00000024
GSADMID	00000026	G*OIMID	00000028	GSDAMID	00000027
GSE MID	00000029	G* MID	0000002A	GSPRMID	0000002B
GSRBT MID	0000002C	G* MID	0000002D	GSVRMID	0000002E
R	00000000	I* CTMID	0000002F	LILOBMID	00000030
MIDMODE	00000005	M* PRI	00000007	MIDROUTE	00000000
MIDSIZE	00000006	M* TYP	00000004	MISSEG	0000000A
MLREQ MID	00000031	M* LCMID	00000032	MULT	00000006
N	00000006	NA G	00000000	NOCAL	00000009
NOGRIZ	R 00000004	NOIGNL	00000002	PACHEAD	FFFFFFFC
PARNBLK	R 00000008	P* T	00000007	PUTPAK	E 00000005
RCB	00000000	RC* A0	00000026	RCBA1	0000002A
RCBA2	0000002E	RC* A3	00000032	RCBA4	00000026
RCBA5	0000003A	RC* A6	0000003E	RCBWI	0000002C
RCBCH2	00000060	RC* CH3	00000064	RCBD0	00000006
RCBDLWK	00000058	RC* LEN	00000067	RCBLNK	00000000
RCBMPG	0000004C	RC* NULL	00000054	RCBPC	00000046
RCBFRY	0000004E	RC* REGS	00000006	RCBSP	00000042
RCBSR	0000004A	RC* STAT	00000052	RCBTIB	00000002
RCBWAIT	00000004	RN :	00000016	ROLLANG	00000005
SICIMID	00000038	S1* 2MID	00000039	SICALMID	00000037
SIMMID	00000033	S1* IMID	0000003A	SIO2MID	0000003B
S1TWMID	0000003C	S2* IMID	0000003E	S2CALMID	0000003D
S2MWMID	00000034	S2* IMID	00000040	S2O2MID	00000041
S2TWMID	00000042	SP* MID	0000003F	SCODSMID	00000035
SCOGSMID	00000036	SCANLMID	00000043	SCCALMID	00000044
SCROMID	00000045	SC* TMID	00000046	SDMID	00000047
STAMID	00000048	TD* TAMID	00000049	TFAMID	0000004B
TLINKMID	0000004A	TO* LNK	00000001	TOTMID	0000004F
TSTRING	R 00000010	UD* IMID	0000004C	UDR2MID	0000004D
UDR3MID	0000004E	WA T	R 00000100		

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.[GR.MC68000_DF]DFA2.LST

17.2 DFA2.LST

ERR LINE ADDR DATE= 08/13/85 TIME= 11:11 38 DFA2

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
  
```

TTL 'DFA2'

VERSION DATE

1.0 7/11/83

1.1 1/16/84 FYW - INSERT TWO TIMING BITS

*SINCLUDE DFA2.FMT/G

MODULE NAME

DFA2.MOD

PROGRAMMER

ETO

DESCRIPTION

DFA2

This job sets up the commands for the DF hardware (ACU, RF processor, DF receiver) and initiates the Fast DF Controller to start the DF measurement. This job is invoked to start a DF or continue on an intermediate step of the current DF. When starting a DF, this job takes the DF parameters for the "new" DF and makes them the "current" DF. The hardware commands are set up, and the Fast DF Controller is initiated. When continuing the current DF, this job simply loads the hardware commands and initiates the Fast DF Controller. To start a new DF (the "next" DF) this job is scheduled by the DF Request Input job (DFA1). When continuing the next step of the current segment, this job is scheduled by the DF Data Input job (DFA3). When this job starts a new DF, it will clear the "done with current" flag to indicate that the DF has started.

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO]IGR.MC68000 DF\DFA2.LST

44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

CALLING SEQUENCE

ROUTED DFA2.#1.

DISC FILES USED

EXTERNAL AND COMMON REFERENCES

HIGHER REFERENCES

DFA1.MOD DF request input (ARF).

LOWERREFERENCES

DETBAND.MOD Determine frequency band

GETBITS.MOD Access BITS fields.

GETLPK.MOD Allocate a large packet.

PUTPAK.MOD Release a small packet for reuse.

CLRACCS.MOD Clear the accumulators for DF data collection.

INITDF.MOD Initialize the DF data collection system.

STARTDF.MOD Initiate (or buffer) a DF collection step.

WAITDF.MOD Wait for the current DF step to complete.

FMTRQ.MOD Format a frequency (5 bytes to 4 bytes).

SIOLOAD.MOD Load the SIO output buffer.

SIOSEND.MOD Prepare SIO for output with FDEC.

REPSETUP.MOD Load a sequence of RF Processor commands.

ACUSETUP.MOD Set up the PIO with a sequence of ACU commands.

NOQUEUE.MOD Enqueue items to identify the DF step.

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

91  *
92  *
93  *
94  *
95  *
96  *
97  *
98  *
99  *
100  *
101  *
102  *
103  *
104  *
105  *
106  *
107  *
108  *
109  *
110  *
111  *
112  *
113  *
114  *
115  *
116  *
117  *
118  *
119  *
120  *
121  *
122  *
123  *
124  *
125  *
126  *
127  *
128  *
129  *
130  *
131  *
132  *
133  *
134  *
135  *
136  *
137  *

DOQUEUE MOD      Dequeue items that identify the DF step.

ADJUSTDF MOD      Adjust FDFC command for bandwidth.

EMPTYFIFO MOD      Empty FDFC fifo into a buffer.

ACQDF MOD         Accumulate DF data from a buffer.

ACQUM MOD         Accumulate DF baseline data.

SIGNPRES MOD      Determine if signal present from DF data.

REANAQC MOD       Read AGC value from DF receiver.

I/O DEVICES USED:
FAST DF CONTROLLER
SIO - 3 CHANNEL A      RF PROCESSOR
SIO - 3 CHANNEL B      DUAL CHANNEL (DF) RECEIVER
PIO                    ANTENNA CONTROL UNITS

ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
ADFUC

METHOD:

* PBL:
* ?DLINK IPCK      /* GET DF REQUEST PACKET */
* DFERR <- NONE    /* INITIALIZE ERROR FLAG */
* TUNED <- FALSE   /* INITIALIZE TUNED RCVR FLAG */
* CALIB <- FALSE    /* INITIALIZE CALIB 1ST SEGMENT FLAG */
* ADFBT <- NOT ADF BITE /* INITIALIZE ADF BITE FLAG */
* SIGNAL <- NOT PRESENT /* INITIALIZE SIGNAL PRESENT FLAG */
* IPACK <- NULL     /* INDICATE NO RESULTS PACKET */
* GET DF TYPE      /* FROM IPCK */
* IF TYPE = BITE
* THEN /* CHECK FOR ADF BITE */
* IF SUBTYPE = COARSE /* CHECK FOR ADF BITE */
* THEN /* DETERMINE IF ON, OFF, OR NORMAL */
* CALL GETBITS(SUBTYPE,DF_SEG) /* GET SEGMENT NO. */
* IF SEG NO. = ADF BITE ON
* THEN /* INDICATE ADF BITE ON */
* PTR <- ADF BITE ON SEQ. HEADER /* GET SEQUENCE HEADER */
* ADFBT <- ON/OFF
* ELSE /* CHECK IF ADF BITE OFF */

```


Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA2.LST

```

138 * IF SEG NO. = ADF BITE OFF
139 * THEN /* INDICATE ADF BITE OFF */
140 * PTR <- ADF BITE OFF SEQ. HEADER
141 * ADFBT <- ON/OFF
142 * ELSE /* NOT ON OR OFF, NORMAL DATA COL. BITE */
143 * ADFBT <- NORMAL
144 * GO TO FINDPTR /* NEED TO GET SEQ. HEADER */
145 *
146 * ENDIF
147 *
148 * ELSE /* TYPE <> BITE */
149 * FINDPTR: CALL DETBAND(DFREQ,BAND) /* DETERMINE FREQ BAND */
150 * CALL GETLPACK(IPACK) /* GET A PACKET FOR RESULTS */
151 * FORMAT PACKET WITH DF IDENTIFIERS
152 * GET POINTER TO BASE OF ACCUMULATORS IN PACKET
153 * CALL CLRACCS(ACCPTR) /* CLEAR OUT ACCUMULATORS */
154 * IF COARSE DF REQUEST
155 * THEN /* CHECK IF ADF BITE */
156 * IF ADFBT = NORMAL
157 * THEN /* GET PSUEDO SEG. NO. FOR ADF BITE */
158 * SEGNO <- ADF BITE SEQ. NO.
159 * ELSE /* NORMAL COARSE DF REQUEST */
160 * SEGNO <- COARSE DF SEG. NO. /* PSUEDO FOR COARSE */
161 *
162 * ENDIF
163 * ELSE /* FINE REQUEST */
164 * CALL GETBITS(DF_SEG,SEGNO) /* GET ACTUAL SEGMENT NO. */
165 * SIGNAL <- TRUE /* ASSUME SIGNAL PRESENT FOR FINE */
166 * IF TYPE = CALIB
167 * THEN /* CHECK IF CALIB SEGMENT NO. 1 */
168 * IF SEGNO = CALIB SEGMENT NO. 1
169 * THEN /* SET FLAG */
170 * CALIB <- TRUE
171 *
172 * ENDIF
173 *
174 * ENDIF
175 * BASED ON BAND AND SEGNO, GET PTR TO SEQUENCE HEADER
176 *
177 * ENDIF
178 * CALL INITDF /* INITIALIZE FDFC */
179 * GET NUM /* FROM DATA COL. SEQ. HEADER */
180 * GET RFPFTR /* FROM DATA COL. SEQ. HEADER */
181 * GET ACUPTR /* FROM DATA COL. SEQ. HEADER */
182 * GET FDFCPT /* FROM DATA COL. SEQ. HEADER */
183 * IF NEED TO TUNE RECEIVER /* CHECK SUBTYPE */
184 * THEN /* GET POINTERS TO TUNING COMMAND SEQUENCES */
185 * TRFPFTR <- TUNING SEQ. RF PROC COMMANDS /* BASED ON BAND */
186 * TACUPTR <- TUNING SEQ. ACU COMMANDS /* BASED ON BAND */
187 * TDFCPT <- TUNING SEQ. FDFC COMMAND /* BASED ON BAND */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.LST

```

185 * CALL FMTRQ(DF_FREQ,OUTFREQ) /* REFORMAT FREQUENCY */
186 * CALL SIOLOAD(OUTFREQ) /* DUMP AGC */
187 * CALL SIOSEND(CHANEL B) /* ENABLE CHANNEL B OUTPUT */
188 * CALL RFPSETUP(RFPPTR) /* LOAD RF PROC. COMMANDS */
189 * CALL SIOSEND(CHANEL A) /* ENABLE CHANNEL A OUTPUT */
190 * CALL ACUSETUP(ACUPTTR) /* LOAD ACU COMMANDS */
191 * CALL STARTDF(TDFPTR) /* INITIATE TUNING STEP */
192 * CALL GETBITS(FDFCCOM,FD_MODE,MODE) /* GET MODE */
193 * COUNT <- 1 /* INITIALIZE STEP COUNT */
194 * CALL NQUEUE(RFPPTR,COUNT,MODE) /* QUEUE ITEMS */
195 * TUNED <- TRUE /* TUNED RCVR FLAG */
196 * IF FINE DF /* CHECK SUBTYPE */
197 * THEN /* SKIP OVER 1ST STEP */
198 * RFPPTR <- RFPPTR + <STEP SIZE> /* JUMP TO NEXT STEP */
199 * ACUPTTR <- ACUPTTR + <STEP SIZE> /* JUMP TO NEXT STEP */
200 * FDFCPTTR <- FDFCPTTR + <STEP SIZE> /* JUMP TO NEXT STEP */
201 *
202 * ENDIF
203 * ELSE /* DO NOT NEED TO TUNE THE RECEIVER */
204 * CALL RFPSETUP(RFPPTR) /* LOAD RF PROC. COMMANDS FOR 1ST STEP */
205 * CALL SIOSEND(CHANEL A) /* ENABLE CHANNEL A OUTPUT */
206 * CALL ACUSETUP(ACUPTTR) /* LOAD ACU COMMANDS FOR 1ST STEP */
207 * CALL STARTDF(FDFCPTTR) /* INITIATE 1ST COLLECTION STEP */
208 * CALL GETBITS(FDFCCOM,FD_MODE,MODE) /* GET MODE OUT OF FDFC COM. */
209 * COUNT <- 1 /* INITIALIZE COUNT */
210 * CALL NQUEUE(RFPPTR,COUNT,MODE) /* QUEUE ITEMS */
211 *
212 * ENDF
213 * CALL GETBITS(DF_BAND,BNDWTH) /* GET BANDWIDTH CODE */
214 * CALL PUTPAK(INK) /* RETURN INPUT PACKET */
215 *
216 * IF COUNT=NUM THEN GO TO DONE /* ONLY ONE STEP */
217 *
218 * LOOP I = 2 TO NUM /* LOOP THROUGH REMAINING STEPS */
219 * CALL RFPSETUP(RFPPTR) /* LOAD A STEP OF RF PROC. COMMANDS */
220 * CALL ACUSETUP(ACUPTTR) /* LOAD A STEP OF ACU COMMANDS */
221 * ENLOOP
222 * CALL SIOSEND(RFPPTR) /* ENABLE RF PROC. COMMAND OUTPUT */
223 * ABRT <- FALSE /* RESET NO SIGNAL ABORT FLAG */
224 *
225 * -----
226 * DF COLLECTION LOOP
227 *
228 * DFLOOP : COUNT <- COUNT + 1 /* INCR. STEP COUNT */
229 * IF COUNT > NUM THEN GO TO DONE /* EXIT LOOP TEST */
230 * GET NEXT FDFC COMMAND /* FROM FDFCPTTR */
231 * CALL ADJFDFC(NFDFC) /* ADJUST FDFC COMMAND */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFA2.LST

```

232 * CALL STARTDF(NDFDC) /* BUFFER THE FDFC COMMAND */
233 *
234 * CALL GETBITS(FD.MODE,MODE) /* GET DATA COLLECTION MODE */
235 * CALL NQUEUE(RFPTR,COUNT,MODE) /* ENQUEUE THE ITEMS */
236 * RFPTR <- RFPTR + <SIZE OF COMMAND STEP>
237 *
238 *
239 * CALL WAITDF /* WAIT FOR FIRST DF */
240 *
241 * CALL DQUEUE(PRPTR,PCOUNT,PMODE) /* GET ASSOCIATED PARAMS.*/
242 * IF DID NOT TUNE RCVR /* CHECK TUNED FLAG */
243 * THEN
244 * IF NOT CALIB SEGMENT NO. 1 /* CHECK CALIB FLAG */
245 * THEN
246 * IF ABRT = FALSE /* SIGNAL IS PRESENT */
247 * THEN
248 * IF SIGNAL NOT PRESENT /* CHECK SIGNAL FLAG */
249 * THEN /* CHECK FOR SIGNAL PRESENCE */
250 * CALL DMFFIFO(BUFFER) /* EMPTY DATA FIFO INTO WORK AREA
251 * CALL SIGPRES(BUFFER,PRPTR,OPPTR,N) /* CHECK FOR
252 * SIGNAL PRESENCE */
253 * IF SIGNAL PRESENT /* CHECK SIGNAL FLAG AGAIN */
254 * THEN /* GET THE GOOD DATA */
255 * LOOP I = 1 TO N
256 * CALL ACCBUF(ACCUMS,OPPTR,OPPTR) /* ACCUMULATE
257 * ENDLOOP
258 * ELSE /* NO NEED TO COLLECT MORE DATA */
259 * ABRT <- TRUE /* SET NO SIGNAL FLAG */
260 *
261 * ENDIF
262 * ELSE /* SIGNAL IS PRESENT */
263 * GET N /* FROM PRPTR */
264 * LOOP I = 1 TO N
265 * CALL ACCUM(ACCUMS,PRPTR,PMODE) /* ACCUMULATE DATA
266 * ENDLOOP
267 * ENDIF /* SIGNAL IS PRESENT */
268 * ELSE /* ABRT = TRUE */
269 * CALL DMFFIFO(BUFFER) /* THROW OUT DATA */
270 * ENDIF /* ABRT = FALSE */
271 * ELSE /* CALIB SEGMENT NO. 1 */
272 * IF PCOUNT = STEP X
273 * THEN /* TAKE AGC MEASUREMENT, NOISE */
274 * CALL READAGC(AGCVAL) /* READ AGC */
275 * SD_SSM <- AGCVAL /* STORE INTO CURRENT DF */
276 * ELSE /* CHECK IF NEED TO TAKE 2ND MEASUREMENT */
277 * IF PCOUNT = STEP Y
278 * THEN /* TAKE AGC MEASUREMENT, SIGNAL */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

279      * CALL READACC(AGCVAL) /* READ AGC */
280      * SD_SSM <- AGCVAL - SD_SSM /* COMPUTE DIFFERENCE */
281      *
282      * ENDIF
283      * ENDIF /* PCOUNT = STEP X */
284      * ENDIF /* CALIB SEGMENT NO. 1 */
285      * ELSE /* DID NOT TUNE RCVR */
286      * CALL DMPIFO(BUFFER) /* THROW DATA AWAY */
287      * TUNED <- FALSE /* RESET FLAG */
288      * ENDIF
289      *
290      * BOTTOM : GO TO DELOOP /* END OF DF LOOP */
291      *
292      * -----
293      * END OF DF COLLECTION LOOP
294      *
295      * -----
296      *
297      * DONE : CALL WAITDF /* EMPTY PIPE */
298      *
299      * CALL DQUEUE(PRFPTR,PCOUNT,PMODE) /* DEQUEUE LAST SET OF ID */
300      * IF NOT ADF BITE ON OR OFF
301      * THEN /* CHECK IF SIGNAL PRESENT */
302      * IF SIGNAL IS PRESENT /* CHECK SIGNAL FLAG */
303      * THEN /* ACCUMULATE DATA */
304      * CALL ACCUM(ACCUMS,PRFPTR,PMODE) /* ACCUMULATE DATA */
305      * ELSE /* FLAG ERROR */
306      * DFERR <- NO SIGNAL
307      *
308      * ENDIF
309      * SD_ERR <- DFERR /* PUT DF ERRORS IN CURRENT DF */
310      * IF FINE DF SEGMENT /* CHECK SUBTYPE */
311      * THEN /* SCHEDULE THE ACCUMULATE FOR FINE JOB */
312      * ?SCHED DFA5,#2,...,IPACK
313      * ELSE /* SCHEDULE THE LOB CALC. JOB */
314      * ?SCHED DFA6,#2,...,IPACK
315      *
316      * ENDIF
317      *
318      * DONE WITH CURRENT <- TRUE /* SET FLAG TO INDICATE DONE */
319      *
320      * ?EXIT
321      *
322      * -----
323      *
324      * DFA2 IDNT -M,-X
325      * OPT XDEF DFA2E

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000 DF]DFA2.LST

326	XDEF	NPEND	
327	XDEF	NCOMPL	
328	XDEF	RESET	
329	XDEF	RESOVF	
330	XDEF	WAITING	
331	XDEF	DFBOX	
332	XDEF	RMSG	
333	XDEF	RID	
334	XDEF	ADATA,TDATA	
335	XDEF	SENDID	
336	XDEF	BAND	
337	XREF	DFDIB	
338	XDEF	RFPLDPOS	
339	XDEF	RFPPTR	
340	XDEF	ENDWDTH	
341	XDEF	ADFRT	
342	XDEF	SIGNAL	
343	XDEF	SKFRST	
344	XDEF	DMCUR,DFERR,NAVDATA	
345	*		
346	XDEF	INTCPU	
347	*		
348	*		
349	*	SUBROUTINES	
350	*		
351	XREF	GETRITS	
352	XREF	DETBAND	
353	XREF	GETLPAK,PUTPAK	
354	XREF	CLRACCS	
355	XREF	INITDF,STARTDF,WAITDF	
356	XREF	FMTFRQ	
357	XREF	SIOLOAD,SIOSEND	
358	XREF	RFSETUP,ACUSETUP	
359	XREF	NQUEUE,DQUEUE	
360	XREF	ADJDFC	
361	XREF	DMPIFO,ACCBUF,ACCUM	
362	XREF	SIGPRES	
363	XREF	DETSIG	
364	XREF	READAGC	
365	*		
366	*	EXTERNAL VARIABLES (OR TABLES)	
367	*		
368	XREF	ABTON,ABTOFF	
369	XREF	GEOTAB	NORMAL SEQUENCE TABLE
370	XREF	SECTAB2	SEQUENCE TABLE FOR COARSE VERIFY
371	XREF	TUNSEQ,TUNEACU,TUNEFOFC	
372	XREF	DFA5,DFA6	

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

373 XREF DFA2DAT1 ;ADDR OF TIMING WORD #1
374 XREF DFA2OUT1 ;ADDR TO OUTPUT TIMING WORD #1 TO
375 XREF DFA2MSK1 ;MASK USED TO SET/CLEAR TIMING BIT #1
376 XREF DFA2DAT2 ;ADDR OF TIMING WORD #2
377 XREF DFA2OUT2 ;ADDR TO OUTPUT TIMING WORD #2 TO
378 XREF DFA2MSK2 ;MASK USED TO SET/CLEAR TIMING BIT #2
379
380 XREF NAVIATA CURRENT NAVIATA
381 XREF CALIB CALIB SEG. 1 FLAG
382
383 *SINCLUDE ARTEMA'S.S/G NOLIST.?DLINK.?SCHED.?EXIT
384 :SPART NOLIST
385 LIST
386
387 *SINCLUDE ARTEDATA.S/G NOLIST.RCB
388 :SPART NOLIST
389 LIST
390
391 *SINCLUDE STRC.MAC/S
392 LIST
393
394 *SINCLUDE DFC.DBS/G
395
396 DFC DF DATA COLLECTION SEQUENCE HEADER.
397
398 EACH HEADER BLOCK CONTAINS THE POINTERS TO
399 THE APPROPRIATE COMMAND SEQUENCE FOR THE
400 RF PROCESSOR, ACU'S, AND FAST DF CONTROLLER.
401
402 DFC_NUM EQU 0 NUMBER OF STEPS IN SEQUE
403 DFC_RFP EQU 2 PTR TO RF PROCESSOR COM
404 DFC_ACU EQU 6 PTR TO ACU COMMANDS
405 DFC_PDFC EQU 10 PTR TO PDFC COMMANDS
406 DFC EQU 14
407
408 *SINCLUDE DF.PAC/G
409
410 DF DF REQUEST (GDPU -> ADPU)
411
412 All DF requests to the ARFs.
413
414 1) DF_FR1 - DF_FR5 FREQUENCY OF DF
415 Frequency consists of 10 BCD digits. The most
416 significant digit represents 1 GHZ.
417
418 2) DF_GAIN RECEIVER GAIN SETTINGS
419 This field is ignored by ADPU software.
420

```

Improved GUARDRAIL V MC68000 'DF' Files DRCL:[ALGO.IGR.MC68000 DF]DFA2.LST

```

694 * 3) DF_BAND BANDWIDTH VALUES
695 * 0 = 8 KHZ
696 * 1 = 15 KHZ
697 * 2 = 50 KHZ
698 *
699 * 4) DF_DET DETECTOR VALUES
700 * 0 = FM
701 * 2 = AM
702 * 4 = CW
703 * 5 = LSB
704 * 6 = USB
705 * 7 = LSB
706 *
707 * 5) DF_TC AGC TIME CONSTANT
708 * This field is ignored by ADPU software
709 *
710 * 6) DF_DIJ AGC DIMP CONSTANT
711 * This field is ignored by ADPU software.
712 *
713 * 7) DF_SQ SQUELCH
714 * This field is ignored by ADPU software.
715 *
716 * 8) DF_CV COARSE VERIFY FLAG
717 * This flag is used only during calibration verify.
718 * 0 = fine DF
719 * 1 = coarse DF
720 *
721 * 9) DF_POL ANTENNA POLARIZATION
722 * This field is ignored by ADPU software.
723 *
724 * 10) DF_TYP DF TYPE DESIGNATOR
725 * 1 = BITE
726 * 2 = MANUAL DF
727 * 3 = INTEROF
728 * 4 = AUTO DF (DS)
729 * 5 = GENERAL SEARCH (FINE)
730 * 6 = CALIBRATION
731 * 7 = AOI
732 * 8 = GEOSCREEN
733 *
734 * 11) DF_AUDIO AUDIO CORRELATOR RECEIVER TUNING
735 * 0 = do not tune audio correlator receiver
736 * 1 = tune audio correlator receiver
737 *
738 * 12) DF_FINE COARSE/FINE DF FLAG
739 * 0 = coarse DF
740

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

741	*	1 = fine DF	
742	*		
743	*	14) DF RESP RESPONSE FLAG	
744	*	0 = no response	
745	*	1 = send response	
746	*		
747	*	14) DF CONT CONTIGUOUS FLAG	
748	*	0 = non contiguous DF	
749	*	1 = contiguous segments - i.e. do no re-tune receiver	
750	*		
751	*	15) DF SEG (DF SEGMENT NUMBER	
752	*	Valid segment numbers are from 1 thru 6.	
753	*		
754	*		
755	*	15) DF FN1,DF FN2 ENABLE ARFX REPORTING	
756	*	0 = Do not send additional DF data messages (UDR1,UDR2,UDR3)	
757	*	1 = Do send additional DF data messages.	
758		DF MID EQU 0	MESSAGE ID (\$12)
759		DF NIB EQU 1	ARF ID & BLOCK COUNT
760		DF ARF EQU 3<<8+4	ARF ID
761		DF BLK EQU 3<<8+0	BLOCK COUNT
762		DF ACC EQU 2	ACCOUNTABILITY
763		DF FRQ EQU 4	1ST 2 BCD DIGITS
764		DF FR1 EQU 4	2ND 2 BCD DIGITS
765		DF FR2 EQU 5	3RD 2 BCD DIGITS
766		DF FR3 EQU 6	4TH 2 BCD DIGITS
767		DF FR4 EQU 7	LAST 2 BCD DIGITS
768		DF FR5 EQU 8	GAIN TYPE & MANUAL CNTRL
769		DF GAIN EQU 9	GAIN TYPE
770		DF TYPE EQU 1<<8+6	MANUAL GAIN SETTING
771		DF MAN EQU 5<<8+0	
772		DF RCVR EQU 10	
773		DF ENG EQU 10	BANDWIDTH AND DETECTORS
774		DF BAND EQU 2<<8+5	BANDWIDTH CODE SEE NOTE
775		DF DET EQU 2<<8+2	DETECTOR CODE SEE NOTE 3
776		DF TC EQU 0<<8+1	AGC TIME CONSTANT
777		DF DU EQU 0<<8+0	AGC DUMP CONSTANT
778		DF FLAGS EQU 11	MISC FLAGS
779		DF EQ EQU 0<<8+7	SQUELCH
780		DF FN1 EQU 0<<8+5	ENABLE ARF1 REPORTING
781		DF FN2 EQU 0<<8+4	ENABLE ARF2 REPORTING
782		DF CV EQU 0<<8+3	COARSE VERIFY FLAG
783		DF POL EQU 1<<8+1	ANTENNA POLARIZATION
784		DF TYP EQU 12	DF TYPE SEE NOTE 4
785		DF AUB EQU 13	DF SUB TYPE & AUDIO CORR
786		DF AUDIO EQU 0<<8+7	AUDIO CORR RCVR TUNE
787		DF FINE EQU 0<<8+6	COARSE/FINE FLAG

Improved GUARDRAIL V MC68000 'DF' Files
 BRCL:ALGO.IGR.MC68000 DF|DFA2.LST

748 00000005	DF RESP EQU	0<<R+5	RESPOND FLAG
749 00000004	DF CONT EQU	0<<R+4	CONTIGUOUS FLAG
750 00000000	DF SEG EQU	3<<R+0	SEGMENT NUMBER
751 0000000E	DF EQU	14	
752	*SINCLUDE SD PAC 5		
753	* SD BASELINE MEASUREMENTS FOR ONE SEGMENT		
754	* SD MESSAGE ID.		
755	* SD DF TYPE CODE		
756	* SD ACCOUNTABILITY		
757 00000000	SD MID EQU	0	
758 00000001	SD TYP EQU	1	
759 00000002	SD ACC EQU	2	
800 00000004	SD ERR EQU	4	
801 00000006	SD RSM EQU	6	
802 00000007	SD ASUB EQU	7	
803 00000007	SD AUDIO EQU	0<<R+7	AUDIO CORR TUNE
804 00000006	SD FINE EQU	0<<R+6	COARSE/FINE FLAG
805 00000005	SD RESP EQU	0<<R+5	RESPOND FLAG
806 00000004	SD CONT EQU	0<<R+4	CONTIGUOUS FLAG
807 00000000	SD SEG EQU	3<<R+0	SEGMENT NUMBER
808 00000008	SD FR1 EQU	8	1ST BYTE OF FREQ.
809 00000009	SD FR2 EQU	9	2ND BYTE OF FREQ.
810 0000000A	SD FR3 EQU	10	3RD BYTE OF FREQ.
811 0000000B	SD FR4 EQU	11	4TH BYTE OF FREQ.
812 0000000C	SD FR5 EQU	12	5TH BYTE OF FREQ.
813 0000000D	SD BAND EQU	13	FREQ. BAND CODE
814 0000000E	SD BA1D1 EQU	14	BASLINE 1 DIR. INPH
815 00000012	SD BA1DQ EQU	18	BASLINE 1 DIR. QUAD
816 00000016	SD BA1R1 EQU	22	BASLINE 1 REV. INPH
817 0000001A	SD BA1RQ EQU	26	BASLINE 1 REV. QUAD
818 0000001E	SD BA2D1 EQU	30	BASLINE 2 DIR. INPH
819 00000022	SD BA2DQ EQU	34	BASLINE 2 DIR. QUAD
820 00000026	SD BA2R1 EQU	38	BASLINE 2 REV. INPH
821 0000002A	SD BA2RQ EQU	42	BASLINE 2 REV. QUAD
822 0000002E	SD BA3D1 EQU	46	BASLINE 3 DIR. INPH
823 00000032	SD BA3DQ EQU	50	BASLINE 3 DIR. QUAD
824 00000036	SD BA3R1 EQU	54	BASLINE 3 REV. INPH
825 0000003A	SD BA3RQ EQU	58	BASLINE 3 REV. QUAD
826 0000003E	SD BA4D1 EQU	62	BASLINE 4 DIR. INPH
827 00000042	SD BA4DQ EQU	66	BASLINE 4 DIR. QUAD
828 00000046	SD BA4R1 EQU	70	BASLINE 4 REV. INPH
829 0000004A	SD BA4RQ EQU	74	BASLINE 4 REV. QUAD
830 0000004E	SD BA5D1 EQU	78	BASLINE 5 DIR. INPH
831 00000052	SD BA5DQ EQU	82	BASLINE 5 DIR. QUAD
832 00000056	SD BA5R1 EQU	86	BASLINE 5 REV. INPH
833 0000005A	SD BA5RQ EQU	90	BASLINE 5 REV. QUAD
834 0000005E	SD BA6D1 EQU	94	BASLINE 6 DIR. INPH

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000_DF]DFA2.LST

1162 00000010	TO DATA DS L 1	*transmitted data
1164 00000020	SENDID DS L 1	*sender id
1166		
1170 00000024	BAND DS W 1	FREQUENCY BAND CODE
1171 00000026	RFPTR DS W 1	SIO PTR FOR RIO - 3 (A)
1172 00000028	RFPTR DS L 1	RF PROC COMMAND PTR
1173		
1174 00000000	EMPTY EQU 1	*indicates null message queue
1175		
1176		
1177	*df returned message types	
1178 00000000	RESULT EQU 0	*normally returned DF message
1179 00000001	CLEAR EQU 1	*indicates need to dump message queue
1180		
1181	*error returns from startdf and waitdf	
1182		
1183 00000000	NONF EQU 0	*no error
1184 00000001	STOVF EQU 1	*start overflow
1185 00000002	ABORT EQU 2	*aborted by call to resetdf
1186 00000003	ROVF EQU 3	*too many completed results in fifo
1187 00000002	MAXCMP EQU 2	*maximum number of lts allowed to accumulate*
1188		
1189 00000002	ADDFON EQU 2	ARF DF BITE ON FLAG
1190 00000003	ADFOFF EQU 3	ARF DF BITE OFF FLAG
1191 FFFFFFFF	ONOFF EQU -1	ADF BITE ON OR OFF FLAG
1192 00000001	YES EQU 1	
1193 00000006	CAL1 EQU 6	CALIB SEGMENT NO. 1 CODE
1194 00000008	BITESEG EQU 8	ADF BITE SEGMENT NO.
1195 00000007	COARSEG EQU 7	COARSE DF SEGMENT NO.
1196 00000070	BANDSEQ EQU 8*DFC	SIZE OF ONE TABLE
1197 00000001	TRUE EQU 1	
1198		
1199 00000010	TWOMEAS EQU 16	2 RAW DATA MEASMT SETS (A & C)
1200 00000008	ONEMEAS EQU 8	1 RAW DATA MEASMT SET (A & C)
1201		
1202 00000000	STORINIT EQU 50	CHANNEL B START ADDR.
1203 00000000	SIOAINIT EQU 5*00	CHANNEL A START ADDR.
1204		
1205 00000002	STEFX EQU 2	CALIB SEGMENT STEP 1
1206 00000004	STEFY EQU 4	CALIB SEGMENT STEP 2
1207 0000002C 0003	NOILEV DC W 5*3	AGC VALUE AT -112 DBM
1208		
1209 0000000C	RFTUNE EQU 6*2	SIZE OF RFP TUNING SEQ.
1210 00000004	ACUTUNE EQU 2*2	SIZE OF ACU TUNING SEQ.
1211		
1212 0000002E	IPACK DS L 1	PACKET POINTER (SD)
1213 00000032	BNDWTH DS W 1	BANDWIDTH

[illegible]

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

1271 *-----
1272 * INITIALIZE FLAGS
1273 *
1274 *
1275 *
1276 CLR.W DFERR INITIALLY NO ERRORS
1277 CLR.W TUNED RESET TUNED RCVR FLAG
1278 CLR.W CALIB RESET CALIB SEG 1 FLAG
1279 CLR.W ADFBT RESET ADF BITE FLAG
1280 CLR.W SIGNAL RESET SIGNAL PRESENT FLAG
1281 CLR.L IPACK INDICATE NO PACKET
1282 CLR.W SKFRST RESET SKIP MSGWMT FLAG
1283 CLR.W DFERR RESET TOOK TOO LONG FLAG
1284 *
1285 * CHECK FOR ADF BITE
1286 *
1287 MOVE.B DF_TYP(A1),D0 GET DF TYPE
1288 CMP.B #TBITE,D0 CHECK IF BITE
1289 BNE.S FINDPTR NOT BITE, CONTINUE
1290 MOVE.B DF_ASUB(A1),D0 GET SUBTYPE
1291 BIST #DF_FINE,D0 CHECK IF ADF BITE
1292 BNE.S FINDPTR BITE, BUT FINE
1293 *
1294 * ADF BITE, CHECK IF ON, OFF, OR NORMAL
1295 *
1296 MOVE.W #DF_SEG,D1 GET DESCRIPTOR
1297 JSR GETBITS EXTRACT SEGMENT NUMBER
1298 CMP.W #ADFON,D1 CHECK IF ADF BITE ON
1299 BNE.S CHKOFF NOT ON, SEE IF OFF
1300 LEA ABTOW,A2 GET PTR TO ADF BITE ON SEQ
1301 MOVE.W #ONOFF,ADFBT SET FLAG TO ON/OFF
1302 BRA STRTCOL READY TO START
1303 *
1304 CHKOFF CMP.W #ADFOFF,D1 CHECK IF ADF BITE OFF
1305 BNE.S NORMBIT NOT ON OR OFF, NORMAL ADF BITE
1306 LEA ABTOFF,A2 GET PTR TO ADF BITE OFF SEQ
1307 MOVE.W #ONOFF,ADFBT SET FLAG TO ON/OFF
1308 BRA STRTCOL READY TO START
1309 *
1310 NORMBIT MOVE.W #YES,ADFBT NORMAL ADF BITE
1311 *
1312 *****
1313 * GET POINTER TO APPROPRIATE DATA COLLECTION SEQUENCE
1314 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

*****
*
1315 FINDPTR LEA DF_FR1(A1),A0 GET PTR OF FREQ STRING
1316 JSR DETBAND DETERMINE FREQUENCY BAND
1317 MOVE.W D0,BAND SAVE BAND CODE
1318 MOVE.W D0,D2 SAVE BAND
1319 MOVE.B DF_FRNG(A1),D0 GET OVERALL FIELD
1320 MOVE.W #DF_BAND,D1 GET DESCRIPTOR
1321 JSR GETBITS GET BAND WIDTH CODE
1322 MOVE.W D1,BNDWDTH SAVE BANDWIDTH CODE
1323
1324 * PREPARE A LARGE PACKET FOR RESULTS
1325
1326
1327
1328 JSR GETLPK GET A LARGE PACKET
1329 MOVE.B D2,SD_BAND(A6) SET FREQ BAND CODE
1330 MOVE.B DF_TYP(A1),SD_TYP(A6) DF TYPE
1331 MOVE.W DF_ACC(A1),SD_ACC(A6) ACCOUNTABILITY
1332 MOVE.B DF_ASUB(A1),SD_ASUB(A6) SUBTYPE
1333 MOVE.B DF_FR1(A1),SD_FR1(A6) 1ST BYTE OF FREQ
1334 MOVE.B DF_FR2(A1),SD_FR2(A6) 2ND BYTE OF FREQ
1335 MOVE.B DF_FR3(A1),SD_FR3(A6) 3RD BYTE OF FREQ
1336 MOVE.B DF_FR4(A1),SD_FR4(A6) 4TH BYTE OF FREQ
1337 MOVE.B DF_FR5(A1),SD_FR5(A6) 5TH BYTE OF FREQ
1338 CLR.W SD_ERR(A6) INITIALIZE ERROR CODE (NONE)
1339 MOVE.L A6,IPACK SAVE PACKET ADDRESS
1340 LEA SD_BA1D1(A6),A0 GET ADDR. OF ACCUMULATORS
1341 MOVE.L A0,ACCBASE SET ACCUMULATOR ADDR.
1342 JSR CLRACCS CLEAR ACCUMULATORS
1343
1344 * GET SEGMENT NUMBER (OR PSEUDO SEGMENT NUMBER)
1345
1346 MOVE.B DF_ASUB(A1),D1 GET SUBTYPE
1347 BTST #DF_FINE,D1 CHECK IF COARSE OR FINE
1348 BNE.S FINE IF 1 THEN FINE
1349 TST.W ADFBT CHECK IF ADF BITE (NORMAL)
1350 BEQ.S COARSE NOT ADF BITE, CONTINUE
1351 MOVE.W #BITESG,D2 SET INDEX TO BITE SEGMENT
1352 BRA.S GETSEQ GO GET SEQUENCE PTR
1353
1354 * COARSE
1355 MOVE.W #COARSEG,D2 SET INDEX TO COARSE SEGMENT
1356 BRA.S GETSEQ GO GET SEQUENCE PTR
1357
1358 * FINE
1359 MOVE.B DF_ASUB(A1),D0 GET OVERALL FIELD
1360 MOVE.W #DF_SEG,D1 GET DESCRIPTOR
1361 JSR GETBITS GET SEGMENT NUMBER
1362 MOVE.W D1,D2 SET SEGMENT NUMBER
1363 MOVE.W #TRUE,SIGNAL ASSUME SIGNAL PRESENT FOR FINE

```

Improved GUARDRAIL V MC68000 'DF' Files DRCL: [ALGO.IGR.MC68000 DF]DFA2.LST

```

000C
1362 0000024A 0C29 0006 000C      CMP.B  #TCALIB,DF_TYP(A1) CHECK IF CALIB
1363 00000250 680E                BNE.S  GETSEQ      IF NOT, CONTINUE
1364 00000252 0C41 0006      CMP.W  #CAL1,D1      CHECK IF CALIB SEG. NO. 1
1365 00000256 6808                BNE.S  GETSEQ      IF NOT, CONTINUE
1366 00000258 33FC 0001 0000 E      MOVE.W  #TRUE,CALIB      ELSE SET FLAG
0000

1367
1368
1369
1370
1371 00000260 303A F0C2      GETSEQ      MOVE.W  BAND,D0      GET BAND CODE
1372 00000264 5340      SUBQ.W  #1,D0      ADJUST FOR INDEX
1373 00000266 C1FC 0070      MULS   #BANDSEQ,D0      MULTIPLY BY TABLE ENTRY
1374 0000026A 45F9 0000 0000 E      LEA    SEQTAB,A2      GET ADDRESS OF SEQUENCE TABLE
1375 00000270 0C29 0006 000C      CMP.B  #TCALIB,DF_TYP(A1) CHECK IF DOING A CALIB
1376 00000276 6610      BNE.S  USETAB1      NOT CALIB, USE ORIGINAL TABLE
1377 00000278 1229 000B      MOVE.B  DF_FLAGS(A1),D1      GET MISC. FLAG FIELD
1378 0000027C 0801 0003      BTST   #DF_CV,D1      CHECK IF COARSE VERIFY
1379 00000280 6706      BEQ.S  USETAB1      NOT COARSE VERIFY, USE OR. TABLE
1380 00000282 45F9 0000 0000 E      LEA    SEQTAB2,A2      COARSE VERIFY, USE OTHER TABLE
1381
1382 00000288 D4C0      USETAB1      ADD.W  D0,A2      ADJUST BY OFFSET
1383 0000028A 5342      SUBQ.W  #1,D2      ADJUST INDEX FOR OFFSET
1384 0000028C C5FC 000E      MULS   #DFC,D2      MULTIPLY BY SUB-ENTRY
1385 00000290 D4C2      ADD.W  D2,A2      GET PTR TO SEQUENCE
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395 00000292 4EB9 0000 0000 E      STRICOL      JSR    INITDF      RESET THE FDFC
1396 00000298 3C2A 0000      MOVE.W  DFC_NUM(A2),D6      GET NUMBER OF STEPS
1397 0000029C 6700 0412      BEQ    FINIO      IF NOTHING TO DO, EXIT
1398 000002A0 2C6A 0092      MOVE.L  DFC_RFP(A2),A6      GET RFP COMMAND PTR
1399 000002A4 2A6A 0006      MOVE.L  DFC_ACU(A2),A5      GET ACU COMMAND PTR
1400 000002A8 286A 000A      MOVE.L  DFC_FDFC(A2),A4      GET FDFC COMMAND PTR
1401
1402
1403
1404 000002AC 1029 000D      * DETERMINE IF NEED TO TUNE RCVR OR NOT
1405 000002B0 0800 0004      *      MOVE.B  DF_ASUB(A1),D0      GET SUBTYPE
1406 000002B4 6600 0118      *      BTST   #DFC_CONT,D0      CHECK IF NEED TO TUNE
                                *      BNE    NOTUNE      IF CONTIGUOUS, NO TUNE

```


Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000_DF]DFA2.LST

```

1407
1408
1409
1410 000002B8 2649
1411 000002BA 43FA FD78
1412 000002BE 41EB 0004
1413 000002C2 4EB9 0000 0000 E
1414 000002C8 4279 0000 003C R
1415 000002CE 102B 0004
1416 000002D2 0240 00F0
1417 000002D6 6708
1418 000002D8 33FC 0001 0000 R
      003C
1419 000002E0 102B 000A
1420 000002E4 08C0 0001
1421 000002E8 0880 0000
1422
1423
1424
1425
1426
1427
1428
1429
1430 000002EC 0280 FFFF FFE3
1431 000002F2 33C0 0000 003A R
1432 000002F8 41FA FD46
1433 000002FC 303C 0006
1434 00000300 4840
1435 00000302 4241
1436 00000304 4EB9 0000 0000 E
1437
1438 0000030A 303C 0000
1439 0000030E 4EB9 0000 0000 E
1440
1441
1442
1443 00000314 41F9 0000 0000 E
1444 0000031A 323A FD08
1445 0000031E 5341
1446 00000320 3401
1447 00000322 C3FC 000C
1448 00000326 D0C1
1449 00000328 23C8 0000 0028 R
1450 0000032E 2448
1451 00000330 4279 0000 0026 R
1452 00000336 4EB9 0000 0000 E

* SET UP TUNING STRINGS FOR DF RECEIVER
*
*
      MOVE.L A1,A3      SAVE A COPY OF IPACK
      LEA   FREQUF,A1    GET PTR TO OUTPUT BUFFER
      LEA   DF_FR1(A3),A0 GET PTR TO BCD STRING
      JSR   FMTFRQ       FORMAT 5 BCD TO 4 BCD
      CLR.W GHZ          CLEAR GHZ FLAG
      MOVE.B DF_FR1(A3),D0 GET 1ST BYTE OF FREQ
      AND.W #SF0,D0      MASK OFF GHZ NIBBLE
      BEQ.S NOGHZ        IF 0 THEN NO GHZ
      MOVE.W #1,GHZ      ELSE SET GHZ BIT

      NOGHZ      MOVE.B DF_RNG(A3),D0  GET BANDWIDTH AND DETECTOR
      BSET #DF_TC,D0  REQ. SLOW AGC TIME CONST.
      BCLR #DF_DU,D0  REQ. AGC DUMP
      1422
;
; NOTE: THE RECEIVER WILL GENERATE A TONE AT EVERY FREQUENCY
; IF THE DETECTOR TYPE SELECTED IS CW. THE TONE IS FOR
; AUDIO MONITORING OF SIGNALS. HOWEVER THE TONE WILL
; CAUSE INCORRECT DETECTIONS OF SIGNALS (I.E. MISINTERPRETING
; THE TONE AS A POSSIBLE SIGNAL). FORCE THE DETECTOR
; TYPE TO FM (CODE = 000) TO PREVENT GENERATION OF THE TONE.
;
      ANDI.L $FFFFFFE3,D0  force detector type to FM
      MOVE.W D0,BW        SET IN DATA BUFFER
      LEA   FREQUF,A0     GET PTR TO FORMAT LIST
      MOVE.W #NFREQ,D0    GET NUMBER OF ELEMENTS
      SWAP D0              GET NUMBER IN UPPER WORD
      CLR.W D1             CLEAR SIO PTR
      JSR   SIOLOAD        LOAD 1ST COMMAND (DUMP AGC)
      *
      MOVE.W #SIOBINIT,D0 GET START ADDR. FOR RCVR CHANNEL
      JSR   SIOSEND        PREPARE SIO FOR OUTPUT
      *
* SET UP RF PROC. COMMANDS FOR TUNING
*
      LEA   TUNSEQ,A0     GET PTR TO RFP TUNING SEQ
      MOVE.W BAND,D1      GET BAND CODE
      SUBQ.W #1,D1        DECR. FOR OFFSET
      MOVE.W D1,D2        GET A COPY
      Muls #RFPTRUNE,D1   MULTIPLY BY SIZE OF ENTRY
      ADD.W D1,A0          ADJUST ADDR. BY OFFSET
      MOVE.L A0,RFPTR     SET RFPTR TO TUNING SEQ
      MOVE.L A0,A2        GET A COPY OF RFPTR
      CLR.W RFPDPOS       RESET SIO ADDR. PTR
      JSR   RFPSETUP      LOAD SIO WITH RFP COMMANDS

```

Improved GUARDRAIL V MC68000 'DF' Files DRG1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

1453 0000033C 303C 0000 GET START ADDR. FOR RFP CHANNEL
1454 00000340 4EB9 0000 PREPARE SIO FOR OUTPUT
1455
1456 *
1457 * SET ACU COMMANDS FOR TUNING
1458
1458 00000346 3202 GET INDEX
1459 00000348 C3FC 0004 MULTIPLY BY SIZE OF ENTRY
1460 0000034C 41F9 0000 GET ADDR. OF ACU TUNING TABLE
1461 00000352 D1C1 ADJUST ADD. BY OFFSET
1462 00000354 4EB9 0000 LOAD PIO WITH ACU COMMANDS
1463
1464 *
1465 * GET FDFC COMMAND FOR TUNING SEQUENCE
1466
1466 0000035A E542 MULTIPLY INDEX BY OFFSET (4)
1467 0000035C 41F9 0000 GET TABLE POINTER
1468 00000362 2030 0000 GET FDFC COMMAND
1469 00000366 4EB9 0000 ADJUST FDFC COMMAND
1470
1471 *
1472 * INITIATE TUNING SEQUENCE
1473
1473 0000036C 4EB9 0000 START DF SEQUENCE
1474
1475 00000372 48E7 80C0 SAVE REGISTERS USED FOR TUNING
1476 00000376 2079 0000 ADDR OF TIMING WORD #2
1477 0000037C 2279 0000 ADDR TO OUTPUT TIMING WORD #2 TO
1478 00000382 3039 0000 MASK USED TO CLEAR TIMING BIT #2
1479 00000388 B150 CLEAR TIMING BIT #2
1480 0000038A 3290 OUTPUT TIMING WORD #2
1481 0000038C 4CDF 0301 RESTORE REGISTERS USED FOR TUNING
1482
1483 *
1484 * IF TUNED THE RCVR FOR A FINE SEGMENT.
1485 * SKIP OVER FIRST COLLECTION STEP
1486 * (TUNED INSTEAD OF 1ST STEP)
1487
1488 00000390 204A GET RFPPTR
1489 00000392 4840 GET FDFC COMMAND
1490 00000394 325C 0104 GET DESCRIPTOR
1491 00000398 4EB9 0000 GET MODE CODE
1492 0000039E 303C 0001 SET STEP NO. TO 1
1493 000003A2 4EB9 0000 QUEUE THE ITEMS
1494 000003A8 3E00 SET STEP COUNT
1495 000003AA 33FC 0001 INDICATE TUNED RCVR
1496
1497 000003B2 102B 0000 GET SUBTYPE
1498 000003B6 0800 0006 CHECK IF FINE OR COARSE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.LST

1499	000003BA	6700	0034	BEQ	DFLOAD	IF COARSE, GO ON
1500	000003BE	301E		MOVE.W	(A6)+,D0	GET NUMBER IN STEP
1501	000003C0	E340		ASL.W	#1,D0	ADJUST BY SIZE FOR OFFSET
1502	000003C2	DCC0		ADD.W	D0,A6	JUMP TO NEXT STEP
1503						
1504	000003C4	301D		MOVE.W	(A5)+,D0	GET NUMBER OF ACU COMMANDS
1505	000003C6	E340		ASL.W	#1,D0	ADJUST BY SIZE FOR OFFSET
1506	000003C8	DAC0		ADD.W	D0,A5	JUMP TO NEXT STEP
1507						
1508	000003CA	588C		ADD.L	#4,A4	JUMP TO NEXT FDPC COMMAND
1509						
1510	000003CC	6072		BRA.S	DFLOAD	
1511						
1512						
1513						
1514						
1515						
1516						
1517	000003CE	4279	0000	NOTUNE	CLR.W	RFPDPOS
1518	000003D4	2649		MOVE.L	A1,A3	RESET SIO LOAD POSITION
1519	000003D6	23CE	0000	MOVE.L	A6,RFPFTR	GET PACKET PTR IN A3
1520	000003DC	4EB9	0000	JSR	RFPSETUP	GET POINTER TO RFP SEQUENCE
1521	000003E2	303C	0800	MOVE.W	#SIOAINIT,D0	LOAD SIO WITH RFP COMMANDS
1522	000003E6	4EB9	0000	JSR	SIOSEND	GET START ADDR. FOR RFP CHANNEL
1523						PREPARE SIO FOR OUTPUT
1524						
1525						
1526	000003EC	204D		MOVE.L	A5,A0	GET PTR TO ACU COMMANDS
1527	000003EE	4EB9	0000	JSR	ACUSETUP	LOAD PIO WITH ACU COMMANDS
1528	000003F4	2A48		MOVE.L	A0,A5	UPDATE ACU PTR
1529						
1530						
1531						
1532	000003F6	201C		MOVE.L	(A4)+,D0	GET FDPC COMMAND
1533	000003F8	4EB9	0000	JSR	ADJFDPC	ADJUST FDPC COMMAND
1534	000003FE	4EB9	0000	JSR	STARTDF	START SEQUENCE
1535						
1536	00000404	48E7	80C0	MOVEM.L	D0/A0-A1,-(SP)	SAVE REGISTERS USED FOR TIMING
1537	00000408	2079	0000	MOVEA.L	(DFA2DAT2),A0	ADDR OF TIMING WORD #2
1538	0000040E	2279	0000	MOVEA.L	(DFA2OUT2),A1	ADDR TO OUTPUT TIMING WORD #2 TO
1539	00000414	3039	0000	MOVE.W	(DFA2MSK2),D0	MASK USED TO CLEAR TIMING BIT #2
1540	0000041A	B150		EOB.W	D0,(A0)	CLEAR TIMING BIT #2
1541	0000041C	3290		MOVE.W	(A0),(A1)	OUTPUT TIMING WORD #2
1542	0000041E	4CDF	0301	MOVEM.L	(SP)+,D0/A0-A1	RESTORE REGISTERS USED FOR TIMING
1543						
1544	00000422	4840		SWAP	D0	GET FDPC COMMAND
1545	00000424	323C	0104	MOVE.W	#FD_MODE,D1	GET DESCRIPTOR

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

1546 00000428 4EB9 0000 0000 E      JSR      GETBITS      GET MODE CODE
1547 0000042E 303C 0001      MOVE.W #1,D0      SET COUNT TO 1
1548 00000432 204E      MOVE.L A6,A0      GET OLD RFPTR
1549 00000434 4EB9 0000 0000 E      JSR      NQUEUE      QUEUE ITEMS
1550
1551 0000043A 2C7A FFFC      MOVE.L RFPTR,A6      UPDATE RFPTR
1552 0000043E 3E00      MOVE.W D0,D7      UPDATE COUNTER
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565 00000440 204E      DFLD      MOVE.L A6,A0      GET A COPY OF RFPTR
1566 00000442 230E 0000 0028 R      MOVE.L A6,RFPTR      PASS PTR TO RFP COMMAND LIST
1567 00000448 2C4B      MOVE.L A3,A6      RETURN PACKET (INPUT) PTR
1568 0000044A 122B 000D      MOVE.B DF,ASUB(A3),D1  GET SUBTYPE FOR LATER TEST
1569 0000044E 4EB9 0000 0000 E      JSR      PUTPAK      RETURN INPUT PACKET
1570 00000454 2C48      MOVE.L A0,A6      RESTORE RFPTR
1571 00000456 BE46      CMP.W D6,D7      CHECK IF DONE YET
1572 00000458 6700 01FC      BEQ      DONE      IF COUNT = TOTAL, DONE
1573
1574 0000045C 3006      MOVE.W D6,D0      GET A COPY OF NO. OF STEPS
1575 0000045E 5340      SUBQ.W #1,D0      ALREADY DONE 1ST STEP
1576 00000460 5340      SUBQ.W #1,D0      DECR. FOR COUNTER
1577 00000462 204D      MOVE.L A5,A0      GET ACU PTR
1578
1579
1580
1581
1582 00000464 4EB9 0000 0000 E      LOADLP      JSR      RFPSETUP      LOAD ONE STEP OF REP COMMANDS
1583 0000046A 4EB9 0000 0000 E      JSR      ACUSETUP      LOAD ONE STEP OF ACU COMMANDS
1584 00000470 51C8 FFF2      DBF      D0,LOADLP      ENDOOP
1585
1586 00000474 23CC 0000 004C R      MOVE.L A4,FDFCOM      PUT FDFC PTR INTO TEMP
1587
1588 0000047A 4279 0000 0010 R      CLR.W ABRT      RESET ABORT FLAG
1589 00000480 4A79 0000 000E R      TST.W PIPE      CHECK IF PIPELINE MODE
1590 00000486 674C 4E71      BEQ      WTDF      NOT PIPE, GO WAIT
1591
1592

```

* 1ST STEP STARTED, LOAD UP THE REST

* OF THE SEGMENT'S COMMANDS (RF PROC. AND ACU).

* AT THIS POINT:

* A6 = RF PROCESSOR COMMAND PTR

* A5 = ACU COMMAND PTR

* A4 = FDFC COMMAND PTR

* LOAD UP ALL THE REST OF THE RF PROC. COMMANDS AND

* ACU COMMANDS FOR THE CURRENT SEGMENT

[illegible]

276

```

DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

	JSR	DQDFUE					DEQUEUE ITEMS
1540	000004DA	4EB9	0000	0000	E		CHECK IF JUST TUNED RCVR
1541	000004E0	4A79	0000	0008	R		DID NOT TUNE, CHECK CALIB
1542	000004E6	6700	000C				CLEAR TUNED RCVR FLAG
1543	000004EA	4279	0000	0008	R		CHECK IF SIGNAL PRESENT (FINE DF)
1544	000004F0	4A79	0000	000C	R		GO GET E BL DATA
1545	000004F6	6600	0000		*		
1546							
	CLR.W	D2					INITIALIZE COUNTER OF GOOD SETS
1547	000004FA	4242					GET NUMBER OF RAW DATA SETS
1548	000004FC	3010					SET A COPY OF NUMBER
1549	000004FE	3600					GET PTR TO TEMP. BUFFER
1550	00000500	45FA FB12					EMPTY FOF INTO A BUFFER
1551	00000504	4EB9	0000	0000	E		
1552					*		
1553	0000050A	4BEA	0010				GET PTR TO BLA DIR. MSRCNT
1554	0000050E	4EB9	0000	0000	E		CHECK IF SIGNAL PRESENT
1555	00000514	4A40					CHECK STATUS OF TEST
1556	00000516	6704					NO SIG, CHECK BL A REV
1557	00000518	343C	0001				SIGNAL PRES., INDICATE
1558	0000051C	50BD			TAREV		SET PTR TO BL A REV
1559	0000051E	4EB9	0000	0000	E		CHECK IF SIGNAL PRESENT
1560	00000524	4A40					CHECK STATUS FLAG
1561	00000526	6604			BNE.S	ADDNUM	IF SIGNAL PRES, UPDATE COUNTER
1562	00000528	4242			CLR.W	D2	NOT TWO IN A ROW
1563	0000052A	6008			BRA.S	TEBL	GO CHECK BL E
1564	0000052C	5242			ADDQ.W	#1,D2	UPDATE COUNTER
1565	0000052E	0C42	0002		CHP.W	#2,D2	CHECK IF FIRST TWO HAD SIGNAL
1566	00000532	6C24			BGE.S	GOODSIG	YES, GO ACCUMULATE E BL
1567					*		
1568	00000534	50BD			TEBL		
1569	00000536	4EB9	0000	0000	E		SET PTR TO BL E RAW DATA SET
1570	0000053C	D440			ADD.W	D0,D2	CHECK IF SIGNAL PRESENT
1571	0000053E	0C42	0002		CHP.W	#2,D2	UPDATE NUMBER BASED ON STATUS
1572	00000542	6734			BFG.S	GOODSGNE	CHECK IF TWO IN A ROW
1573	00000544	0C42	0001		CHP.W	#1,D2	SIGNAL GOOD, BUT DON'T ACC. BL E
1574	00000548	6600	00FC		BNE	BOTTOM	CHECK IF ONLY BLE WAS GOOD
1575	0000054C	33FC	0001	0000	R		NO SIGNAL, DON'T USE DATA
1576	00000554	6000	00F0				ONLY BL E HAD SIGNAL
1577					*		
1578	00000558	33FC	0001	0000	R		SIGNAL PRESENT
1579	00000560	2848			MOVE.L	A0,A4	SET RFP COMMAND PTR
1580	00000562	264A			MOVE.L	A2,A3	SET DATA PTR
1581	00000564	267A FA5A			MOVE.L	ACCBASE,A0	GET PTR TO ACCUMULATORS
1582	00000568	5343			SUBQ.W	#1,D3	DECR. NUMBER FOR INDEX
1583	0000056A	4EB9	0000	0000	E		UNLOAD FROM BUFFER
1584	00000570	51CB FF5B			DBF	D3,TUNEBUF	ENDLOOP

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000 DFJDF.A2.LST

```

1685 00000574 6000 0000 BRA BOTTOM
1686
1687 00000578 33FC 0001 0000 R GOODSIGNE MOVE.W #TRUE,SIGNAL SET SIGNAL PRESENT FLAG
000C
1688 00000580 6000 0000 BRA BOTTOM
1689
1690
1691 00000584 4A79 0000 0000 E TCAL TST.W CALIB CHECK IF CAL SEG. NO. 1
1692 0000058A 6662 BNE.S CALIB1 IF YES, GO PROCESS
1693
1694 0000058C 4A79 0000 0000 R TST.W SIGNAL CHECK IF SIGNAL PRESENT
1695 00000592 6644 BNE.S GETDATA IF YES, GO GET DATA
1696
1697
1698
1699
1700
1701 00000594 4A79 0000 0010 R TST.W ABRT CHECK IF NO SIGNAL ALREADY FOUND
1702 0000059A 6600 009E BNE EMPFIF UNLOAD DATA
1703
1704 0000059E 45FA FAR4 LEA BUFFER,A2 ELSE, CHECK FOR SIGNAL PRESENCE
1705 000005A2 3010 MOVE.W (A0),D0 GET NUMBER OF MEASUREMENTS
1706 000005A4 4EB9 0000 0000 E JSR DMFFIFO UNLOAD FIFO INTO BUFFER
1707
1708 000005AA 2248 MOVE.L A0,A1 GET PTR TO RFP SEQUENCE
1709 000005AC 4EB9 0000 0000 E JSR SIGPRES LOOK FOR SIGNAL PRESENCE
1710 000005B2 4A79 0000 0000 R TST.W SIGNAL SEE IF SIGNAL PRESENT
1711 000005B8 660C BNE.S ACCFB IF PRESENT, GO COLLECT
1712 000005BA 33FC 0001 0000 R MOVE.W #TRUE,ABRT ELSE SET ABORT FLAG
0010
1713 000005C2 6000 0000 ACCFB BRA BOTTOM ELSE, ACCUMULATE GOOD DATA
1714 000005C6 5343 MOVE.L ACCBASE,A0 GET PTR TO ACCUMULATORS
1715 000005C8 207A FAR6 MOVE.L ACCBUF UNLOAD FROM BUFFER
1716 000005CC 4EB9 0000 0000 E JSR D3,BUFLOOP ENDOOP
1717 000005D2 51CB FFF8 DBF D3,BUFLOOP
1718 000005D6 606E BRA.S BOTTOM
1719
1720
1721
1722 000005D8 2248 GETDATA MOVE.L A0,A1 GET RFP COMMAND PTR
1723 000005DA 207A FAR4 MOVE.L ACCBASE,A0 GET PTR TO ACCUMULATORS
1724 000005DE 3619 MOVE.W (A1)+,D3 GET COUNT
1725 000005E0 5343 SUBQ.W #1,D3 DECR. FOR COUNTER
1726 000005E2 4EB9 0000 0000 E JSR ACCUM ACCUMULATE FROM FIFO
1727 000005E8 51CB FFF8 DBF D3,FIFLOOP ENDOOP
1728 000005EC 6058 BRA.S BOTTOM
1729

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1: {ALGO.IGR.MC68000 DFJ.DFA2.LST

```

1730 * CALIBRATION SEQUENCE NO. 1,
1731 * CHECK IF NEED TO TAKE AGC READINGS
1732 *
1733 CALIB1 CMP.W #STEPX,D0 CHECK IF 1ST READ
1734 BEQ.S FRSTREAD IF YES GO D0 IT
1735 CMP.W #STREAP,D0 CHECK IF 2ND READ
1736 BEQ.S SECREAD IF YES, GO D0 IT
1737 BRA.S BOTTOM ELSE, LOOP BACK
1738 *
1739 FRSTREAD JSR READAGC READ AGC VALUE OF NOISE
1740 MOVE.L IPACK,A2 GET PACKET ADDR.
1741 AND.W #S00FF,D0 ONLY WANT LS BYTE
1742 CMP.W NOILEV,D0 CHECK IF IN < -112 RANGE
1743 BGT.S STREAD1 GREATER THAN, GO STORE
1744 MOVE.W NOILEV,D0 ELSE LESS THAN, SET TO NOISE
1745 MOVE.B D0,SD_SSM(A2) PUT NOISE MEASURE IN
1746 BRA.S BOTTOM
1747 *
1748 SECREAD JSR READAGC READ AGC VALUE OF SIGNAL
1749 MOVE.L IPACK,A2 GET PACKET PTR
1750 MOVE.B SD_SSM(A2),D1 GET AGC VALUE OF NOISE
1751 AND.W #S00FF,D0 ONLY WANT LS BYTE
1752 AND.W #S00FF,D1 ONLY WANT LS BYTE
1753 SUB.W D1,D0 CALC. SIGNAL - NOISE
1754 MOVE.B D0,SD_SSM(A2) PUT INTO PACKET
1755 BRA BOTTOM
1756 *
1757 * * THROW DATA OUT (EMPTY FDFC FIFO)
1758 *
1759 ENFFIF MOVE.W (A0)+,D0 GET COUNT
1760 LEA BUFFER,A2 GET BUFFER ADDR.
1761 JSR DMFFIFO EMPTY FIFO
1762 *
1763 *****
1764 *
1765 * * END OF DF COLLECTION LOOP
1766 *
1767 *****
1768 *
1769 BOTTOM TST.W PIPE CHECK IF PIPELINE MODE
1770 BNE DFLOOP NOT PIPE, GO TO TOP
1771 ADD.W #1,D7 BUMP UP COUNTER
1772 BRA STDF START NEXT ONE
1773 *
1774 * * EMPTY PIPE LINE.
1775 *
1776 DONE JSR WAITDF WAIT FOR DF TO COMPLETE

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000 DF]DFA2.LST

```

1777 *
1778 0000065C 4EB9 0000 0000 E JSR DQUEUE DEQUEUE ITEMS
1779 *
1780 00000662 4A79 0000 000A R TST.W ADFRT CHECK IF ADF BITE ON/OFF
1781 00000668 5800 0006 BMI EXIT IT IS, FINISHED
1782 *
1783 0000066C 4A79 0000 000C R TST.W SIGNAL CHECK IF SIGNAL PRESENT
1784 00000672 6716 BEQ.S NOSIG IF NO SIGNAL, FLAG ERROR
1785 *
1786 00000674 2248 MOVE.L A0,A1 GET RFP COMMAND PTR
1787 00000676 3619 MOVE.W (A1)+,D3 GET COUNT
1788 00000678 207A F906 MOVE.L ACCBASE,A0 GET ACCUMULATORS
1789 0000067C 5343 SUBQ.W #1,D3 DECR. COUNT FOR COUNTER
1790 0000067E 4EB9 0000 0000 E FIFLOOP2 JSR ACCUM ACCUMULATE FROM FIFO
1791 00000684 51CB FFF8 DBF D3,FIFLOOP2 ENDLOOP
1792 00000688 602E BRA.S FINI
1793 *
1794 * NO SIGNAL, FLAG ERROR
1795 *
1796 0000068A 3018 NOSIG MOVE.W (A0)+,D0 GET COUNT TO EMPTY FIFO
1797 0000068C 45FA F906 LEA BUFFER,A2 GET BUFFER PTR
1798 00000690 4EB9 0000 0000 E JSR DMPPFIFO EMPTY FIFO TO BUFFER
1799 *
1800 00000696 4AB9 0000 002E R TST.L IPACK SEE IF THERE IS A RESULTS PACKET
1801 0000069C 5762 BEQ.S EXIT IF NO PACKET THEN DONE
1802 0000069E 2C7A F906 MOVE.L IPACK,A6 GET RESULTS PACKET PTR
1803 000006A2 302E 0004 MOVE.W SD_ERR(A6),D0 GET ERRORS SO FAR
1804 000006A6 08C0 0002 BSET #NOSIGNAL,D0 INDICATE NO SIGNAL PRESENT
1805 000006AA 3D40 0004 MOVE.W D0,SD_ERR(A6) UPDATE ERROR CODE
1806 000006AE 6008 BRA.S FINI
1807 *
1808 * NOTHING TO DO, RETURN INPUT PACKET
1809 *
1810 000006B0 2C49 FINIO MOVE.L A1,A6 RETURN INPUT PACKET
1811 000006B2 4EB9 0000 0000 E JSR PUTPAK
1812 *
1813 * SEE WHAT TO DO WITH RESULTS
1814 *
1815 000006B8 4AB9 0000 002E R FINI TST.L IPACK CHECK IF A PACKET TO PASS
1816 000006BE 6740 BEQ.S EXIT IF NO PACKET, DONE
1817 000006C0 2C7A F906 MOVE.L IPACK,A6 GET PACKET ADDR.
1818 000006C4 51CB FFF8 TST.W DFERR CHECK IF TOOK TOO LONG
1819 000006C8 6008 BEQ.S GOSEND OKAY, GO AHEAD
1820 000006CC 4A79 0000 000C R MOVE.W SD_ERR(A6),D0 GET ERROR CODE SO FAR
1821 000006D0 08C0 0002 BSET #TOOLONG,D0 INDICATE ERROR
1822 000006D4 51CB FFF8 MOVE.W D0,SD_ERR(A6) UPDATE ERROR CODE
1823 000006D8 6008 GOSEND MOVE.B SD_ASUB(A6),D0 GET SURTYPE

```

Improve: GUARDRAIL V MC68000 'DF' Files BRCL: [ALGO.IGR.MC68000 DF]DFA2.LST

	RTST	#SD	FINE	D0	CHECK IF FINE OR COARSE
1824 000006C8 0500 0000	REQ.S	COAREXT			COARSE
1825 000006CC 671A	%SCHED	DFA5, #2, ..., A6			
1826 000006CE	BRA.S	EXIT			
1861 000006E6 6018					
1862					
1863 000006E8	COAREXT	%SCHED	DFA6, #2, ..., A6		
1898					
1899					
1900					
1901					
1902 00000700 2079 0000 0000 E	MOVEA.L	(DFA2DAT1), A0			ADDR OF TIMING WORD #1
1903 00000706 2279 0000 0000 E	MOVEA.L	(DFA2OUT1), A1			ADDR TO OUTPUT TIMING WORD #1 TO
1904 0000070C 3039 0000 0000 E	MOVE.W	(DFA2MSK1), D0			MASK USED TO CLEAR TIMING BIT #1
1905 00000712 B150	EOB.W	D0, (A0)			CLEAR TIMING BIT #1
1906 00000714 3290	MOVE.W	(A0), (A1)			OUTPUT TIMING WORD #1
1907 00000716 4279 0000 0008 R	CLR.W	INTCPU			
1908					
1909 0000071C					
1923					
1924 00000726					

ASSEMBLER ERRORS = 0

SYMBOL TABLE

ABORT	00000002	ABRT	R 00000010	ABTOFF	E 00000018
ABTON	E 00000017	ACCBASE	R 00000050	ACCBUF	E 00000012
ACCFE	R 000005C6	ACCMID	000000C0	ACCRMID	00000001
ACCUM	E 00000013	ACOUNT	R 000000D4	ACUSETUP	E 0000000D
ACUTUNE	00000004	ADNUM	E 0000052C	ADFBT	R 0000000A
ADTOFF	00000003	ADPON	00000002	ADJDFC	E 00000010
ALJLP	R 00000460	ALNTMID	00000002	ALOBMID	00000003
ARFINMID	00000004	ASTAMID	00000005	ASTRMID	00000006
BAND	R 00000024	BANDSEQ	00000070	BANGLMID	00000007
BITEMID	0000000E	BITEMID	00000009	BITESEG	00000008
BITEMID	00000008	BITEMID	0000000A	BLSGA	00000004
BITEMID	00000003	BITEMID	R 00000032	BOTTOM	R 00000046
BITEMID	0000000B	BITEMID	0000000C	BUFFER	R 00000054
BITEMID	0000000C	BITEMID	R 0000003A	CALL	00000006
BITEMID	0000000C	BITEMID	R 000005EE	CALCMID	0000000D
BITEMID	00000077	CALIB1	00000001	CLRACCS	E 00000005
BITEMID	0000001A	CLAR	00000001	CLRACCS	E 00000005
BITEMID	0000001A	CLAR	E 0000022C	COARSEG	00000007
BITEMID	0000001A	CLAR	00000010	CTLMID	00000011
BITEMID	0000001A	CLAR	FFFFF48A	DETBAND	E 00000002
BITEMID	0000001A	CLAR	0000000E	DFA2DAT1	E 00000020
BITEMID	0000001A	CLAR	R 000000DA	DFA2MSK1	E 00000022

AD-A184 057

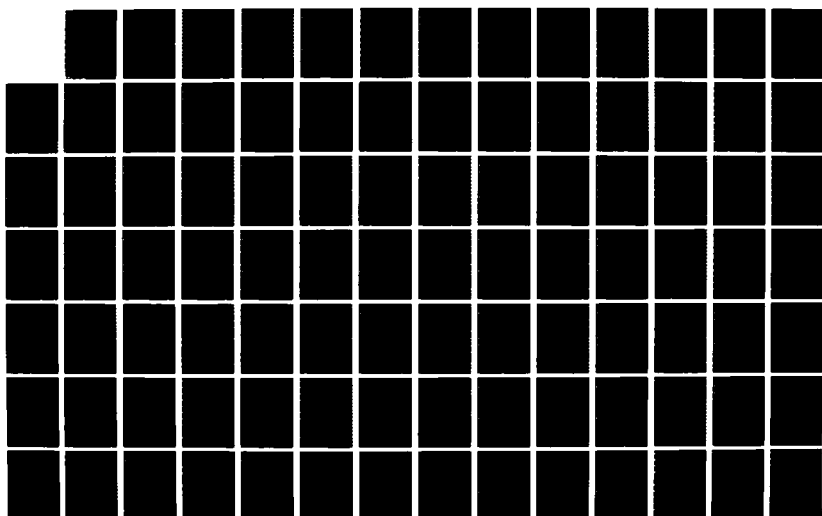
US ARMY INTELLIGENCE CENTER AND SCHOOL (USATCS)
SOFTWARE ANALYSIS AND MAN (U) JET PROPULSION LAB
PASADENA CA 8 PARDO 05 MAR 87 JPL-D-4216 NAS7-918

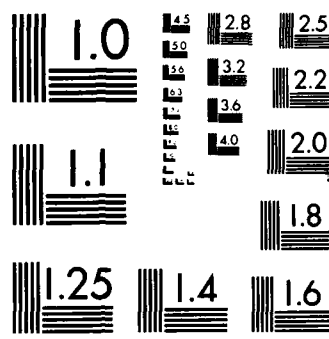
4/5

UNCLASSIFIED

F/G 12/5

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.LST

DFA2MSK2	E	00000025	DFA2OUT1	E	00000021	DFA2OUT2	E	00000024
DFA5	E	0000001E	DFA6	E	0000001F	DFOBOX	R	00000014
DFC	0000000E	DFC_ACU	DFC_FDFC	00000006	DFC_FDFC	DFC_FDFC	0000000A	0000000A
DFC_NUM	00000000	DFC_RFP	DFDIB	00000002	DFDIB	DFDIB	E	00000000
DFDMID	00000013	DFLOAD	DFLOOP	R	00000440	DFLOOP	R	0000048A
DFDMID	00000012	DFPRMID	DFREJ	00000014	DFREJ	DFREJ	00000000	00000000
DFRJMID	00000016	DFRMID	DFACC	00000015	DFACC	DFACC	00000002	00000002
DF_ARF	000000304	DFASUB	DF_AUDIO	0000000D	DF_AUDIO	DF_AUDIO	00000007	00000007
DF_BAND	00000205	DF_BLK	DF_CONT	00000300	DF_CONT	DF_CONT	00000004	00000004
DF_CV	00000003	DF_DET	DF_DU	00000202	DF_DU	DF_DU	00000000	00000000
DF_EN1	00000005	DF_EN2	DF_FINE	00000004	DF_FINE	DF_FINE	00000006	00000006
DF_FLAGS	00000008	DF_FR1	DF_FR2	00000004	DF_FR2	DF_FR2	00000005	00000005
DF_FR3	00000006	DF_FR4	DF_FR5	00000007	DF_FR5	DF_FR5	00000008	00000008
DF_FRQ	00000004	DF_GAIN	DF_MAN	00000009	DF_MAN	DF_MAN	00000500	00000500
DF_MID	00000000	DF_MIB	DF_POL	00000001	DF_POL	DF_POL	00000101	00000101
DF_RCV	0000000A	DF_RESP	DF_RNG	00000005	DF_RNG	DF_RNG	0000000A	0000000A
DF_SEG	00000300	DF_SQ	DF_TC	00000007	DF_TC	DF_TC	00000001	00000001
DF_TYP	0000000C	DF_TYPE	DFMID	00000106	DFMID	DFMID	00000017	00000017
DMFFIFO	E	00000011	DONE	R	00000656	DMID	00000018	00000018
DQUEUE	E	0000000F	DSACKMID	00000019	DSACKMID	DSACKMID	0000001A	0000001A
DSDOMID	0000001B	DSEMD	DSEMD	0000001C	DSEMD	DSEMD	0000001D	0000001D
DSROMID	0000001E	DSPRMID	DSRMID	0000001F	DSRMID	DSRMID	00000020	00000020
EMFFIF	R	0000063A	EMPTY	00000000	EXIT	EXIT	R	00000700
FD	00000002	FDFCOM	FD_ACU	R	0000004C	FD_ACU	0000030C	0000030C
FD_COM	00000000	FD_DLY	FD_MODE	00000102	FD_MODE	FD_MODE	00000104	00000104
FD_NORM	00000000	FD_RCV	FD_RFP	00000106	FD_RFP	FD_RFP	00000308	00000308
FD_TEST	00000001	FIFLOOP	FIFLOOP2	R	000005E2	FIFLOOP2	R	0000067E
FINDPTR	R	0000019A	FINI	R	00000232	FINI	R	000006B8
FINIO	R	00000680	FMTFRQ	E	00000009	FREQBUF	R	00000034
FREQMT	R	00000040	FRSTREAD	R	000005FC	GETBITS	E	00000001
GETDATA	R	00000508	GETLPK	E	00000003	GETSEQ	R	00000260
GHZ	R	0000003C	GINTMID	00000021	GLOBMID	GLOBMID	00000022	00000022
GOODSNE	R	00000578	GOODSIG	R	00000558	GOSEND	R	000006C4
GPROJMID	00000023	GSACKMID	GSACKMID	00000025	GSACKMID	GSACKMID	00000024	00000024
GSAIDMID	00000026	GSAOIMID	GSDAMID	00000028	GSDAMID	GSDAMID	00000027	00000027
GSEMD	00000029	GSMID	GSPRMID	0000002A	GSPRMID	GSPRMID	0000002B	0000002B
GSRTMID	0000002C	GSMID	GSRMID	0000002D	GSRMID	GSRMID	0000002E	0000002E
INITDF	E	00000006	INTCPU	R	000000D8	IPACK	R	0000002E
ISECTMID	0000002F	LLLOMID	LOADLP	00000030	LOADLP	LOADLP	R	00000464
MAXCOMP	00000002	MIDMODE	MIDPRI	00000005	MIDPRI	MIDPRI	00000007	00000007
MIDROUTE	00000000	MIDSIZE	MIDTYP	00000006	MIDTYP	MIDTYP	00000004	00000004
MISSEG	00000008	MLREQMID	MODLCMID	00000031	MODLCMID	MODLCMID	00000032	00000032
MULT	00000006	NARG	NAVDATA	00000000	NAVDATA	NAVDATA	E	00000026
NCOMPL	R	00000002	NFREQ	00000006	NOCAL	NOCAL	00000009	00000009
NOGHZ	R	000002E0	NOILEV	R	0000002C	NONE	00000000	00000000
NORMBIT	R	00000192	NOSIG	R	0000068A	NOSIGNL	00000002	00000002
NOTUNE	R	000003CE	NFEND	R	00000000	NOUEUE	E	0000000E

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFA2.LST

ONEMAS	00000008	ONOFF	FFFFFFF	PCNTL	FFFFF48C
PFIT	00000007	PIPE	R 0000000E	FSTAT	FFFFF48E
PUPAK	E 00000004	RCB	00000000	RCBA0	00000026
RCBA1	0000002A	RCBA2	0000002E	RCBA3	00000032
RCBA4	00000036	RCBA5	0000003A	RCBA6	0000003E
RCBCW1	0000005C	RCBCW2	00000060	RCBCW3	00000064
r 'BD0	00000006	RCBDLWK	00000058	RCBLEN	00000067
RCBLNK	00000000	RCBMPG	0000004C	RCBNUL	00000054
RCBPC	00000046	RCBPRTY	0000004E	RCBREGS	00000006
RCBSP	00000042	RCBSR	0000004A	RCBSTAT	00000052
RCBTIB	00000002	RCBWAIT	00000004	RDATA	R 00000018
READAGC	E 00000016	RESET	R 00000004	RESOVF	R 00000005
RESULT	00000000	RFPLDPOS	R 00000026	RFPTPR	R 00000028
RFPSSETUP	E 0000000C	RFPTUNE	0000000C	RID	R 00000016
RMSG	R 00000016	ROLLANG	00000005	ROVF	00000003
SIC1MID	00000038	S1C2MID	00000039	S1CALMID	00000037
S1MWID	00000033	S1O1MID	0000003A	S1O2MID	0000003B
S1NEMID	0000003C	S2C1MID	0000003E	S2CALMID	0000003D
S2MWID	00000034	S2O1MID	00000040	S2O2MID	00000041
S2NEMID	00000042	SERMID	0000003F	SCODSMID	00000035
SCOGSMID	00000036	SCANLMID	00000043	SCCALMID	00000044
SCROWID	00000045	SCVTMID	00000046	SD	00000096
SDMID	00000047	SD_ACC	00000002	SD_ASUB	00000007
SD_AUDIO	00000007	SD_BA1DI	0000000E	SD_BA1DQ	00000012
SD_BA1RI	00000016	SD_BA1RQ	0000001A	SD_BA2DI	0000001E
SD_BA2DQ	00000022	SD_BA2RI	00000026	SD_BA2RQ	0000002A
SD_BA3DI	0000002E	SD_BA3DQ	00000032	SD_BA3RI	00000036
SD_BA3RQ	0000003A	SD_BA4DI	0000003E	SD_BA4DQ	00000042
SD_BA4RI	00000046	SD_BA4RQ	0000004A	SD_BA5DI	0000004E
SD_BA5DQ	00000052	SD_BA5RI	00000056	SD_BA5RQ	0000005A
SD_BA6DI	0000005E	SD_BA6DQ	00000062	SD_BA6RI	00000066
SD_BA6RQ	0000006A	SD_BA7DI	0000006E	SD_BA7DQ	00000072
SD_BA7RI	00000076	SD_BA7RQ	0000007A	SD_BA8DI	0000007E
SD_BA8DQ	00000082	SD_BA8RI	00000086	SD_BA8RQ	0000008A
SD_BAEI	0000008E	SD_BAEQ	00000092	SD_BAND	0000000D
SD_CONT	00000004	SD_ERR	00000004	SD_FINE	00000006
SD_FR1	00000008	SD_FR2	00000009	SD_FR3	0000000A
SD_FR4	0000000B	SD_FR5	0000000C	SD_MID	00000000
SD_RESP	00000005	SD_SEG	00000030	SD_SSM	00000006
SD_TYP	00000001	SECRETAD	R 0000001A	SENDID	R 00000020
SEQTAB	E 00000019	SEQTAB2	E 0000001A	SIGNAL	R 0000000C
SIGFRES	E 00000014	SIOAINIT	00000080	SIOBINIT	00000000
SIOLOAD	E 0000000A	SIOSEND	E 0000000B	SKFEST	R 00000012
STAMID	00000048	STARTDF	E 00000007	STDF	R 00000042
STEXK	00000002	STEPY	00000004	STOVF	00000001
STREAD1	R 00000014	STRTCOL	R 00000029	TAOI	00000007
TAREV	R 0000001C	TAUTO	00000004	TBITE	00000001

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFA2.LST

TCAL	R	00000584	TCALIB	00000006	TDATA	R	0000001C
TDATANID		00000049	TEBL	R	00000534	TFAMID	0000004B
TGEO		00000008	TGS	00000005	TINTEROP		00000003
TLINCHID		0000004A	TMANUAL	00000002	TOOLNG		00000001
TOTMID		0000004F	TRUE	00000001	TUNEACU	E	0000001C
TUNEBUF	R	0000056A	TUNED	R	00000008	TUNEFDFC	E
TUNESQ	E	0000001B	TWOEAS	00000010	UDRIMID		0000004C
UDR2MID		0000004D	UDR3MID	0000004E	USETAB1	R	00000288
WAITDF	E	00000008	WAITING	R	00000006	WTDF	R
YES		00000001	_FITCT	00000010	_BYTCT		00000002
_CURMD		00000002	_OBYTCT	00000000	_TEMP		00000000
_TMP		00000000					

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO:IGR.MC68000_DF]DFA6.LST

17.3 DFA6.LST

ERR LINE ADDR DATE= 05/29/85 TIME= 10:46:09 DFA6 PAGE 1

```

1  *
2  *-----*
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
  
```

```

TTL      'DFA6'

VERSION  DATE
1.0      8/15/83
1.1      1/16/84
1.2      5/30/84
1.3      6/1/84
1.4      6/15/84
1.5      8/20/84
1.6      8/28/84
1.7      8/28/84
1.8      8/28/84
1.9      8/28/84
1.10     8/28/84
1.11     8/28/84
1.12     8/28/84
1.13     8/28/84
1.14     8/28/84
1.15     8/28/84
1.16     8/28/84
1.17     8/28/84
1.18     8/28/84
1.19     8/28/84
1.20     8/28/84
1.21     8/28/84
1.22     8/28/84
1.23     8/28/84
1.24     8/28/84
1.25     8/28/84
1.26     8/28/84
1.27     8/28/84
1.28     8/28/84
1.29     8/28/84
1.30     8/28/84
1.31     8/28/84
1.32     8/28/84
1.33     8/28/84
1.34     8/28/84
1.35     8/28/84
1.36     8/28/84
1.37     8/28/84
1.38     8/28/84
1.39     8/28/84
1.40     8/28/84
1.41     8/28/84
1.42     8/28/84
1.43     8/28/84
  
```

DFA6

This job is scheduled with either the baseline data for one segment, the accumulated data for a Fine DF, or the accumulated data for a Calibration DF (see DFA3,DFA5). This job takes the baseline data and calculates an LOB. If the input was from a coarse DF (one segment), the current Navigation data is read to give an LOP (for a calibration or fine DF, the Navigation data is taken during the middle segment of the data collection segments). If an LOP cannot be calculated, it is reflected in the DF response message.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

44 * CALLING SEQUENCE
45 *
46 *
47 * ?SCHED DFA6,#3,...,IPACK
48 *
49 *
50 *
51 * Int*4/Input
52 * Pointer to packet containing baseline phase
53 * angles. Either packet definition "TFA" or
54 * "CDF".
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *
67 * BCD BIN. MOD Convert BCD to binary integer.
68 *
69 * COARSE. MOD Coarse search of LOP calculation
70 *
71 * FMTDFR. MOD Format DF response message
72 *
73 * P2SRCH. MOD Pass 2 search of LOP calculation.
74 *
75 * PARFIT. MOD Three point parabolic fit.
76 *
77 * PUTPAK. MOD Release a small packet for reuse.
78 *
79 * GETPAK. MOD Allocate a small packet.
80 *
81 * I/O DEVICES USED:
82 *
83 * ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
84 * ADFUC
85 *
86 * METHOD:
87 *
88 * PDL:
89 *
90 * ?DLINK IPACK /* GET INPUT PACKET */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

91 * IF NO RESPONSE /* CHECK IF ADF BITE SET UP */
92 * THEN /* NOTHING TO DO */
93 * GO TO DONE
94 *
95 * ENDIF
96 * CALL GETPAK(OPACK) /* GET A PACKET FOR RESPONSE */
97 * FORMAT PACKET WITH DF IDENTIFICATION ITEMS
98 * GET ERROR CODE /* CHECK IF ANY ERRORS SO FAR */
99 * IF ANY FATAL ERRORS
100 * THEN /* SEND RESPONSE */
101 * GO TO SNDRESP
102 *
103 * ENDIF
104 * ADFBIT <- FALSE /* INITIALIZE ARF DF BITE FLAG */
105 * IF ARF DF BITE /* CHECK IF RESULTS FOR ARF DF BITE */
106 * THEN
107 * ADFBIT <- TRUE /* SET FLAG */
108 * ELSE /* OTHERWISE CHECK SIGNAL PRESENCE */
109 * CALL BLSGPRS(BASelines,BAND,BADL,STATUS)
110 * IF STATUS = ABORT
111 * THEN /* INDICATE ERROR AND SEND RESPONSE */
112 * ERROR CODE <- ERROR CODE + NO SIGNAL ERROR
113 * EXIT TO SNDRESP
114 *
115 * ENDIF
116 * IF STATUS = NO SIGNAL PRESENT
117 * THEN /* INDICATE WARNING */
118 * ERROR CODE <- ERROR CODE + NO SIGNAL WARNING
119 *
120 * ENDIF
121 * IF SUBTYPE = COARSE
122 * THEN
123 * GET NAV DATA FROM CURRENT NAV DATA
124 * ELSE /* FINE DF */
125 * GET NAV DATA FROM INPUT PACKET /* TFA */
126 *
127 * ENDIF
128 * IF ROLL ANGLE > MAXROLL /* CHECK ROLL ANGLE */
129 * THEN /* INDICATE ERROR */
130 * ERROR CODE <- ERROR CODE + ROLL ANGLE ERROR
131 * EXIT TO SNDRESP /* FATAL ERROR */
132 *
133 * ENDIF
134 * GET NUMBER OF BASELINES /* BASED ON FREQ. BAND */
135 * CALL INQUAD(BASE,OBUFFER1) /* CONVERT DATA TO ANGLES */
136 * CALL DIRREV(OBUFFER1,DFR BASE) /* COMPUTE SINGLE ANGLE PER BL */
137 * CALL BCDBIN(FREQ,INTFRQ) /* COMPUTE INTEGER FREQUENCY */
138 * IF ADFBIT = FALSE
139 * THEN /* TRY TO CALCULATE LOB */
140 * CALL COARSE(THETA,INTFRQ,STATUS) /* THETA = PTR TO BASELINE
141 * ANGLES */
142 * IF STATUS = BAD

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

138 * THEN /* FAILED THE COARSE SEARCH */
139 * ERROR CODE <- ERROR CODE + MULT. MINIMA /* INDICATE ERROR */
140 * EXIT TO SDRRESP /* FATAL ERROR */
141 *
142 * ENDIF
143 * CALL BCDIN(FREQ,IFROMHZ) /* IFROMHZ <- FREQ IN MHZ */
144 * CALL SQTHRESH(QTHRESH) /* COMPUTE Q THRESHOLD */
145 * CALL P2SRCH /* PERFORM 2ND PASS SEARCH */
146 * IF STATUS = BAD
147 * THEN /* FAILED 2ND PASS SEARCH */
148 * ERROR CODE <- ERROR CODE + POOR FIT
149 * ENDIF
150 * PUT LOB AND QUALITY INTO DFR PACKET
151 * ELSE /* ARF DF BITE */
152 * RESP <- ALL FAIL /* INITIALIZE RESULTS */
153 * I <- 0 /* INITIALIZE INDEX */
154 * LOOP 6 TIMES /* NUMBER OF ARF DF BITE FREQ.S */
155 * IF IFRQ = FRQLST(I)
156 * THEN /* FOUND WHICH ARF DF BITE FREQ */
157 * EXIT LOOP TO BTFRND
158 * ENDIF
159 * I <- I + 1 /* INCR. INDEX */
160 * ENDOLOOP
161 * GO TO UPDRS /* FREQ NOT FOUND, REPORT ALL FAIL */
162 * BTFRND: LOOP K=1 TO NBL /* NUMBER OF BASELINES */
163 * DIFF <- | DFR_BASE(K) - TEL(I,K) | /* COMPUTE DIFFERENCE */
164 * IF DIFF < DELTA /* CHECK IF OKAY DIFFERENCE */
165 * THEN /* BASELINE PASSES */
166 * RESP(K) <- PASS
167 * ENDIF
168 * ENDOLOOP
169 * UPDRS: ERROR CODE <- RESP /* RETURN RESULTS */
170 * ENDIF
171 * SDRRESP: ?SCHED CPUSEND,#4,,,OPACK
172 *
173 * DONE: CALL PUTPAK(IPACK) /* RETURN INPUT PACKET */
174 * ?EXIT
175 *
176 * -----
177 *
178 * DFA6 IDNT
179 * OPT -M
180 *
181 * XDEF DFACE
182 *
183 * * PARAMETERS ASSOCIATED WITH CURRENT DF DATA
184

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFA6.LST

```

185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231

*
XDEF THETA
XDEF INTFRQ
XDEF BADBL
XDEF NBL
XDEF MINVAL1
XDEF MINX1
XDEF NORES
XDEF FRLPTR
XDEF FRHPTX
XDEF IFRQMHZ
XDEF BANDLB
XDEF ROLL
XDEF MINVAL2
XDEF MINX2
XDEF LOB
XDEF QUAL
XDEF QTHRESH
XDEF NORES2
XDEF NAVDATA
XDEF MAXROLL
XDEF ERROR2
XDEF SCMIN, SCVAL

*
* TIMING PARAMETERS
*
XREF DFA6DAT1
XREF DFA6OUT1
XREF DFA6MSK1
XREF DFA6DAT2
XREF DFA6OUT2
XREF DFA6MSK2
XREF DFA6DAT3
XREF DFA6OUT3
XREF DFA6MSK3
XREF DFA6DAT4
XREF DFA6OUT4
XREF DFA6MSK4

*
XREF DFRPEN

*
* SUBROUTINES
*
XREF GETPAK
XREF BLSGPRS
XREF INPQUAD
XREF DIRREV

;addr of timing word #1
;addr to output timing word #1
;mask used to set/clear timing bit #1
;addr of timing word #2
;addr to output timing word #2
;mask used to set/clear timing bit #2
;addr of timing word #3
;addr to output timing word #3
;mask used to set/clear timing bit #3
;addr of timing word #4
;addr to output timing word #4
;mask used to set/clear timing bit #4

DF REPORTING ENABLE FLAG

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFA6.LST

232	XREF	BCDBIN	
233	XREF	COARSE	
234	XREF	P2SRCH	
235	XREF	SQTHRESH	
236	XREF	CPUSEND	
237	XREF	PUTLPAK	
238			
239	*	* ARF DF BITE TABLE	
240	*		
241	XREF	BITETBL	
242			
243	*	*\$INCLUDE ARTEMACS.S/G NOLIST,?DLINK,?SCHED,?EXIT	
244	;SPART	NOLIST	
359		LIST	
360	*		
361	*	*\$INCLUDE ARTEDATA.S/G NOLIST,RCB	
362	;SPART	NOLIST	
396		LIST	
397	*		
398	*	*\$INCLUDE STRC.MAC/S	
526		LIST	
527	*		
528	*	*\$INCLUDE TFA.PAC/G	
529	*		
530	*	TFA	BASELINE MEASUREMENTS FOR ONE SEGMENT
531	*		MUST BE SAME AS SD PACKET UP TO TFA_LAT
532	*		
533	TFA MID	EQU	0 MESSAGE ID.
534	TFA_TYP	EQU	1 DF TYPE CODE
535	TFA_ACC	EQU	2 ACCOUNTABILITY
536	TFA_ERR	EQU	4 ERROR CODE
537	TFA_SSM	EQU	6 SIGNAL STRENGTH MEASURE
538	TFA_ASUB	EQU	7 DF SUBTYPE
539	TFA_AUD	EQU	0<<8+7 AUDIO CORR TUNE
540	TFA_FINE	EQU	0<<8+6 COARSE/FINE FLAG
541	TFA_RESP	EQU	0<<8+5 RESPOND FLAG
542	TFA_CONT	EQU	0<<8+4 CONTIGUOUS FLAG
543	TFA_SEG	EQU	3<<8+0 SEGMENT NUMBER
544	TFA_FR1	EQU	8 1ST BYTE OF FREQ.
545	TFA_FR2	EQU	9 2ND BYTE OF FREQ.
546	TFA_FR3	EQU	10 3RD BYTE OF FREQ.
547	TFA_FR4	EQU	11 4TH BYTE OF FREQ.
548	TFA_FR5	EQU	12 5TH BYTE OF FREQ.
549	TFA_BAND	EQU	13 FREQ. BAND CODE
550	TFA_BASE	EQU	14 BASELINE DATA
551	TFA_BAEI	EQU	142 BASELINE E INPH
552	TFA_BAEQ	EQU	146 BASELINE E QUAD

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

553 00000096 TFA_LAT EQU 150 LATITUDE
554 0000009A TFA_LON EQU 154 LONGITUDE
555 0000009E TFA_HEA EQU 158 HEADING
556 000000A0 TFA_ROLL EQU 160 ROLL ANGLE
557 000000A2 TFA EQU 162
558 *
559 *$INCLUDE DFR.PAC/G
560 *
561 * DFR DF RESPONSE (ADPU -> GDFU)
562 *
563 * MESSAGE TO TRANSFER DF RESPONSE INFORMATION
564 *
565 * 1) DFR_ERR DF ERRORS BIT MAP BIT ASSIGNMENTS
566 * 0 = DF REJECTED (PREV. TOO LONG )
567 * 1 = DF DID NOT COMPLETE IN SEGMENT
568 * 2 = NO SIGNAL PRESENT
569 * 3 = BASE LINE SIG PRES FAILED
570 * 4 = BASE LINE SIG PRES ABORT
571 * 5 = ROLL ANGLE EXCEEDS LIMIT
572 * 6 = MULTIPLE MINIMA IN 1ST SEARCH
573 * 7 = POOR FIT, BAD QUALITY
574 * 8 = MISSING SEGMENT (FINE DF)
575 * 9 = NO CAL TABLES
576 *
577 * 2) DFR_TYP DF TYPE VALUES
578 *
579 * 1 = BITE
580 * 2 = MANUAL DF
581 * 3 = INTEROP
582 * 4 = AUTO DF (DS)
583 * 5 = GENERAL SEARCH (FINE)
584 * 6 = CALIBRATION
585 * 7 = AOI
586 * 8 = GEOSCREEN
587 *
588 * 3) DFR_SSM SIGNAL STRENGTH MEASURE
589 * This field contains the digitized AGC voltage which is a
590 * crude measure of signal strength. Division of this field
591 * by 1.5 should give and approximation of SNR in dB. This
592 * field is only valid for calibration measurements.
593 *
594 * 4) DFR_LAT LATITUDE
595 * Latitude of ARF at time of measurement in BAM.
596 *
597 * 5) DFR_LON LONGITUDE
598 * Longitude of ARF at time of measurement in BAM.
599 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618 00000000
619 00000001
620 00000304
621 00000300
622 00000002
623 00000004
624 00000006
625 00000007
626 00000008
627 0000000C
628 00000010
629 00000012
630 00000014
631 00000016
632 00000018
633 00000028
634
635
636
637
638
639
640 00000000
641 00000004
642 00000008
643 0000000A
644 0000000C
645
646

* 6) DFR_LOB LINE OF BEARING
* Line of bearing to emitter with respect to ARF heading.
*
* 7) DFR_QUAL QUALITY OF CAL TABLE FIT
* Quality of the base line measurements cal table fit. Valid
* values are 0 thru 32 with 0 being a perfect fit to the cal
* table.
*
* 8) DFR_HEA HEADING
* Heading of ARF at time of measurement
*
* 9) DFR_ROLL ROLL ANGLE
* Roll angle of ARF at time of measurement
*
* 10) DFR_BASE BASELINE PHASE MEASUREMENTS
* Baseline phase measurements: 5 values for VHF LO, 8
* values for VHF HI and 6 values for UHF.
*
DFR_MID EQU 0 MESSAGE ID ($15)
DFR_NIB EQU 1 ARF ID & BLOCK COUNT
DFR_ARF EQU 3<<8+4 ARF ID
DFR_BLK EQU 3<<8+0 BLOCK COUNT
DFR_ACC EQU 2 ACCOUNTABILITY
DFR_ERR EQU 4 ERROR CODE SEE NOTE 1
DFR_TYP EQU 6 DF TYPE SEE NOTE 2
DFR_SSM EQU 7 SIGNAL STRENGTH MEASURE
DFR_LAT EQU 8 LATITUDE (BAM)
DFR_LON EQU 12 LONGITUDE (BAM)
DFR_LOB EQU 16 LINE OF BEARING (BAM)
DFR_QUAL EQU 18 QUALITY
DFR_HEA EQU 20 HEADING (BAM)
DFR_ROLL EQU 22 AIRCRAFT ROLL (BAM)
DFR_BASE EQU 24 5, 6 OR 8 BASELINE PHASE
DFR EQU 40
*
* $INCLUDE NAV.DRS/G
*
* NAV NAVIGATION DATA BASE
* NAV HOLDING DATA BASE FOR NAVIGATION DATA
*
NAV_LAT EQU 0 LATITUDE
NAV_LONG EQU 4 LONGITUDE
NAV_HEAD EQU 8 HEADING
NAV_ROLL EQU 10 ROLL ANGLE
NAV EQU 12
*
* $INCLUDE UDR1.PAC/G

```

[illegible]

293

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

694 00000028
695
696
697
698
699
700
701
702
703 00000000
704 00000001
705 00000304
706 00000300
707 00000002
708 00000004
709 00000028
710
711
712
713
714
715
716
717
718
719
720
721 00000000
722 00000001
723 00000304
724 00000300
725 00000002
726 00000004
727 00000014
728 00000015
729 00000016
730 00000017
731 00000018
732 0000001A
733 0000001C
734 0000001E
735 00000020
736 00000022
737 00000024
738
739
740

UDR1 EQU 40
*
*$INCLUDE UDR2.PAC/G
*
* UDR2 (ADPU -> GDFU) MESSAGE TO RETURN
* ADDITIONAL DF DATA FOR DF REPORT JOB.
* SECOND OF THREE ADDITIONAL DATA MESSAGES
* (SEE DR1, DR3)
*
UDR2_MID EQU 0 MESSAGE ID ($4D)
UDR2_NIB EQU 1 ARF ID AND BLOCK COUNT
UDR2_ARF EQU 3<<8+4 ARF ID.
UDR2_BLK EQU 3<<8+0 BLOCK COUNT
UDR2_ACC EQU 2 ACCNTBLTY OF DF REQUEST
UDR2_ER2 EQU 4 SECOND PASS ERROR ARRAY
UDR2 EQU 40
*
*$INCLUDE UDR3.PAC/G
*
* UDR3 (ADPU -> GDFU) MESSAGE TO RETURN
* ADDITIONAL DF DATA FOR DF REPORT JOB.
* THIRD OF THREE ADDITIONAL DATA MESSAGES
* (SEE DR1, DR2)
*
* 1) FREQUENCY CONSISTS OF 10 BCD DIGITS. THE MOST
* SIGNIFICANT DIGIT REPRESENTS 1 GHZ.
*
UDR3_MID EQU 0 MESSAGE ID. ($4E)
UDR3_NIB EQU 1 ARF ID AND BLOCK COUNT
UDR3_ARF EQU 3<<8+4 ARF ID.
UDR3_BLK EQU 3<<8+0 BLOCK COUNT
UDR3_ACC EQU 2 ACCNTBLTY OF DF REQUEST
UDR3_ER2 EQU 4 SECOND PASS ERROR ARRAY
UDR3_FR1 EQU 20 FREQUENCY SEE NOTE 1
UDR3_FR2 EQU 21 2ND BYTE OF FREQUENCY
UDR3_FR3 EQU 22 3RD BYTE OF FREQUENCY
UDR3_FR4 EQU 23 4TH BYTE OF FREQUENCY
UDR3_FR5 EQU 24 5TH BYTE OF FREQUENCY
UDR3_BAD EQU 26 SKIPPED BASELINE (VHF HI
UDR3_MN1 EQU 28 MINIMUM FROM 1ST PASS (
UDR3_MV1 EQU 30 MINIMUM FROM 1ST PASS (V
UDR3_SCM EQU 32 SEC. MIN. FROM 1ST PASS
UDR3_SCV EQU 34 SEC. MIN. FROM 1ST PASS
UDR3 EQU 36
*
*$INCLUDE DFERRS.EQU/G
*

```

295

Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

911 00000030	MINVAL1	DS.W	1	MINIMUM VALUE FROM COARSE
912 00000032	MINX1	DS.W	1	MINIMUM INDEX FROM COARSE
913 00000034	NORES	DS.B	1	ERROR CODE FROM COARSE
914 00000036	FRLEPTR	DS.L	1	FREQ LOW CAL TABLE PTR
915 0000003A	FRHPTR	DS.L	1	FREQ. HIGH CAL TABLE PTR
916 0000003E	IFRQMHZ	DS.W	1	FREQ. IN MHZ (INTEGER)
917 00000040	BANDLB	DS.W	1	FREQ BAND CODE
918 00000042	ROLL	DS.W	1	ROLL ANGLE (16 BIT BAM)
919 00000044	MINVAL2	DS.L	1	MINIMUM VALUE FROM 2ND SEARCH
920 00000048	MINX2	DS.W	1	MINIMUM INDEX FROM 2ND SEARCH
921 0000004A	LOB	DS.W	1	LOB CALCULATED IN 2ND SEARCH
922 0000004C	QUAL	DS.L	1	QUALITY OF LOB
923 00000050 0000 0001	QTHRESH	DC.L	1	QUALITY THRESHOLD TEST
924 00000054	NORES2	DS.B	1	ERROR CODE FROM 2ND SEARCH
925	*			
926 00000006	N10KHZ	EQU	6	6 DIGITS FOR 10 KHZ FREQ
927 00000032	RNDFRQ	EQU	50	ROUND TO MHZ CONSTANT
928 00000064	KHZ	EQU	100	MHZ DIVISOR
929	*			
930 00000056 071C	DELTA	DC.W	\$071C	DIFFERENCE (10 DEG. BAM)
931	*			
932 00000058 01	PETTYPE	DC.B	TBITE	BITE
933 00000059 02	DC.B	TMANUAL		MANUAL DF
934 0000005A 06	DC.B	TCALIB		CALIBRATION
935	*			
936 00000002	NPETYPE	EQU	3-1	NUMBER OF PE TYPES - 1
937 00000004	NFREOB	EQU	5-1	NUMBER OF FREQ. BYTES
938 00000008	NDR2ER2	EQU	9-1	NO. OF ERROR2 ITEMS FOR DR2
939 00000003	NDR3ER2	EQU	4-1	NO. OF ERROR2 ITEMS FOR DR3
940	*			
941	*			
942 0000005B 00	NTAB	DC.B	0	NO. OF BL IN VHF LO
943 0000005C 05	DC.B	5		NO. OF BL IN VHF HI
944 0000005D 08	DC.B	8		NO. OF BL IN UHF
945 0000005E 06	DC.B	6		BLOCK COUNT OF 4
946	*			
947 00000004	DFRNIB	EQU	\$04	FOR ALIGNMENT
948	*			
949 00000060	DS.W	0		GET INPUT PACKET
950	*			
951 00000060	DFAGE	?DLINK	A5	
952	*			
953	*			
954 0000006C 2679 0000 0000 E	MOVEA.L	(DFA6DAT1),A3		:ADDR OF TIMING WORD #1
955 00000072 2879 0000 0000 E	MOVEA.L	(DFA6OUT1),A4		:ADDR TO OUTPUT TIMING WORD #1 TO
956 00000078 3C39 0000 0000 E	MOVE.W	(DFA6MSK1),D6		:MASK USED TO SET TIMING BIT #1
957 0000007E 8D53	OR.W	D6,(A3)		:SET TIMING BIT #1
958 00000080 3893	MOVE.W	(A3),(A4)		:OUTPUT TIMING WORD #1

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

969 00000082 102D 0007
970 00000086 0800 0005
971 0000008A 6700 0412
972 0000008A 6700 0412
973
974 0000008E 4EB9 0000 0000 E
975 00000094 1D7C 0015 0000
976 0000009A 1D7C 0004 0001
977 000000A0 3D6D 0002 0002
978 000000A6 3D6D 0004 0004
979 000000AC 1D6D 0001 0006
980 000000B2 1D6D 0006 0007
981
982 000000B8 302D 0004
983 000000BC 0240 0004
984 000000C0 6600 02F6
985
986 000000C4 102D 0001
987 000000C8 4239 0000 0000 R
988 000000CE 0C00 0001
989 000000D2 6616
990 000000D4 102D 0007
991 000000D8 0800 0006
992 000000DC 660C
993 000000DE 13FC 0001 0000 R
994 000000E6 6000 00C6
995
996 000000EA 41ED 000E
997 000000EE 102D 000D
998 000000F2 4880
999 000000F4 4241
1000 000000F6 142D 0007
1001 000000FA 0802 0006
1002 000000FE 6704
1003 00000100 323C 0001
1004
1005 00000104 2679 0000 0000 E
1006 0000010A 2879 0000 0000 E
1007 00000110 3C39 0000 0000 E
1008 00000116 8D53
1009 00000118 3893
1010
1011
1012 0000011A 4EB9 0000 0000 E
1013
1014

-----
*
MOVE.B TFA_ASUB(A5),D0 GET SUBTYPE
BTST #TFA_RESP,D0 CHECK IF NEED TO RESPOND
BEQ DONE DON'T NEED TO RESPOND
*
JSR GETPAK GET A PACKET FOR RESPONSE
MOVE.B #DFRMIID,DFR_MID(A6) PUT IN MESSAGE ID.
MOVE.B #DFRNIB,DFR_NIB(A6) ARF ID. AND BLK CNT
MOVE.W TFA_ACC(A5),DFR_ACC(A6) ACCOUNTABILITY
MOVE.W TFA_ERR(A5),DFR_ERR(A6) ERROR CODE
MOVE.B TFA_TYP(A5),DFR_TYP(A6) DF TYPE
MOVE.B TFA_SSM(A5),DFR_SSM(A6) SIGNAL STRENGTH MEAS.
*
MOVE.W TFA_ERR(A5),D0 GET ERROR CODE
AND.W #FATAL1,D0 CHECK IF ANY FATAL
BNE SNDRESP FATAL ERROR, EXIT
*
MOVE.B TFA_TYP(A5),D0 GET DF TYPE
CLR.B ADFBIT RESET ADF BITE FLAG
CMP.B #TBITE,D0 CHECK IF BITE
BNE.S NADFBIT NOT ADF BITE
MOVE.B TFA_ASUB(A5),D0 GET SUBTYPE
BTST #TFA_FINE,D0 CHECK IF ADF BITE
BNE.S NADFBIT NOT ADF BITE
MOVE.B #TRUE,ADFBIT SET ADF BITE FLAG
BRA GTNUM GO PROC. ADF BITE
*
NADFBIT LEA TFA_BASE(A5),A0 GET PTR TO ACCUMULATORS
MOVE.B TFA_BAND(A5),D0 GET FREQ BAND CODE
EXT.W D0 MAKE A WORD
CLR.W D1 INITIALIZE TO COARSE
MOVE.B TFA_ASUB(A5),D2 GET SUBTYPE
BTST #TFA_FINE,D2 CHECK IF FINE DF
BEQ.S SIGPRS IF COARSE GO TEST DATA
MOVE.W #1,D1 SET FLAG TO FINE DF
*
SIGPRS MOVEA.L (DFA6DAT2),A3 ;ADDR OF TIMING WORD #2
MOVEA.L (DFA6OUT2),A4 ;ADDR TO OUTPUT TIMING WORD #2
MOVE.W (DFA6MSK2),D6 ;MASK USED TO SET TIMING BIT #2
OR.W D6,(A3) ;SET TIMING BIT #2
MOVE.W (A3),(A4) ;OUTPUT TIMING WORD #2
*
JSR BLSGPRS CHK SIG PRES AND BL S
*

```

Improved GUARDRAIL V MC68000 'DF' Files BRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

1015 00000120 BD53      EOR.W D6,(A3)      ;CLEAR TIMING BIT #2
1016 00000122 3893      MOVE.W (A3),(A4)      ;OUTPUT TIMING WORD #2
1017 -----
1018 00000124 0C41 0002      CMP.W #ABORT,D1      CHECK IF FAILED TEST
1019 00000128 6610      BNE.S GTBADBL      NO ABORT
1020 0000012A 302D 0004      MOVE.W TFA_ERR(A5),D0      GET ERROR CODE
1021 0000012E 08C0 0004      BSET #BLSGA,D0      INDICATE FAILURE
1022 00000132 3D40 0004      MOVE.W D0,DFR_ERR(A6)      PUT INTO RESPONSE PACKET
1023 00000136 6000 01F0      BRA SNDRESP
1024 *
1025 0000013A 4A41      GTBADBL      TST.W D1      CHECK IF NO SIGNAL
1026 0000013C 670C      BEQ.S BLSGP      SIGNAL PRESENT
1027 0000013E 302D 0004      MOVE.W TFA_ERR(A5),D0      GET ERROR CODE
1028 00000142 08C0 0003      BSET #BLSGF,D0      INDICATE WARNING
1029 00000146 3D40 0004      MOVE.W D0,DFR_ERR(A6)      UPDATE FIELD
1030 0000014A 33C2 0000 0024 R      MOVE.W D2,BADBL      SET BAD BASELINE
1031 *
1032 00000150 102D 0007      MOVE.B TFA_ASUB(A5),D0      GET SUBTYPE
1033 00000154 0800 0006      BTST #TFA_FINE,D0      CHECK IF FINE OR COARSE
1034 00000158 6620      BNE.S FINENAV      IF FINE, XFER NAV DATA
1035 0000015A 43F9 0000 0000 E      LEA NAVDATA,A1      GET PTR TO DATA BASE
1036 00000160 2D69 0000 0008      MOVE.L NAV_LAT(A1),DFR_LAT(A6) GET LATITUDE
1037 00000166 2D69 0004 000C      MOVE.L NAV_LONG(A1),DFR_LOW(A6) GET LONGITUDE
1038 0000016C 3D69 0008 0014      MOVE.W NAV_HEAD(A1),DFR_HEA(A6) HEADING
1039 00000172 3D69 000A 0016      MOVE.W NAV_ROLL(A1),DFR_ROLL(A6) ROLL ANGLE
1040 00000178 6018      BRA.S TSTROLL      GO TEST ROLL ANGLE
1041 *
1042 0000017A 2D6D 0096 0008      FINENAV      MOVE.L TFA_LAT(A5),DFR_LAT(A6) XFER LATITUDE
1043 00000180 2D6D 000A 000C      MOVE.L TFA_LOW(A5),DFR_LOW(A6) XFER LONGITUDE
1044 00000186 3D6D 000E 0014      MOVE.W TFA_HEA(A5),DFR_HEA(A6) HEADING
1045 0000018C 3D6D 00A0 0016      MOVE.W TFA_ROLL(A5),DFR_ROLL(A6) ROLL ANGLE
1046 *
1047 00000192 302E 0016      TSTROLL      MOVE.W DFR_ROLL(A6),D0      GET ROLL ANGLE
1048 00000196 6A02      BPL.S POSROLL      IF ROLL IS POSITIVE, OKAY
1049 00000198 4440      NEG.W D0      ELSE CONVERT TO POSITIVE
1050 0000019A B079 0000 0000 E      CMP.W MAXROLL,D0      CHECK IF TOO LARGE
1051 000001A0 6F0C      BLE.S GTNUM      IF LESS, THEN OKAY
1052 000001A2 302E 0004      MOVE.W DFR_ERR(A6),D0      GET ERROR CODE
1053 000001A6 08C0 0005      BSET #ROLLANG,D0      INDICATE ROLL ANGLE
1054 000001AA 3D40 0004      MOVE.W D0,DFR_ERR(A6)      UPDATE FIELD
1055 *
1056 000001AE 102D 000D      GTNUM      MOVE.B TFA_BAND(A5),D0      GET FREQ. BAND CODE
1057 000001B2 4880      EXT.W D0      MAKE A WORD
1058 000001B4 33C0 0000 0040 R      MOVE.W D0,BANDLB      SAVE BAND CODE
1059 000001BA 43FA FE9F      LEA NTAB,A1      GET PTR TO TABLE OF NUM.S
1060 000001BE 1A31 0000      MOVE.B 0(A1,D0.W),D5      GET NUMBER OF BASELINES
1061 000001C2 4885      EXT.W D5      MAKE A WORD

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

1062	000001C4	33C5	0000	002A	R	MOVE.W	D5,NBL	SAVE NUMBER
1063								
1064	000001CA	43FA	FF36			LEA	BUFFER,A1	GET PTR TO TEMP BUFFER
1065	000001CE	204D				MOVE.L	A5,A0	GET PTR TO INPUT PACKET
1066	000001D0	4EB9	0000	0000	E	JSR	INPUAD	CONVERT TO DIR & REV. ANGLES
1067								
1068	000001D6	45EE	0018			LEA	DFR_BASE(A6),A2	GET PTR TO FINAL BUFFER
1069	000001DA	23CA	0000	002C	R	MOVE.L	A2,THETA	SAVE PTR TO FINAL ANGLES
1070	000001E0	4EB9	0000	0000	E	JSR	DIRREV	COMBINE DIR., REV. RCVR PHASE
1071								
1072	000001E6	302E	0004			MOVE.W	DFR_ERR(A6),D0	GET ERROR CODE SO FAR
1073	000001EA	0240	0024			AND.W	#ERRMSK,D0	CHECK FOR FATAL ERRORS
1074	000001EE	6500	0138			BNE	SNDRESP	IF PRESENT, GO SEND IT
1075								
1076	000001F2	41ED	0008			LEA	TFA_FRI(A5),A0	GET PTR TO BCD FREQ
1077	000001F6	303C	000A			MOVE.W	#NFREQ,D0	GET NUMBER OF DIGITS
1078	000001FA	4EB9	0000	0000	E	JSR	BCDBIN	CONVERT TO INTEGER FREQ.
1079	00000200	23C1	0000	0026	R	MOVE.L	D1,INTFREQ	SAVE INT. FREQ.
1080								
1081	00000206	4A39	0000	0000	R	TST.B	ADFBIT	CHECK IF ADF BITE
1082	0000020C	6500	00C2			BNE	ADFBITP	GO PROCESS ADF BITE (NO LOB)
1083								
1084								
1085	00000210	2678	0000	0000	E	MOVEA.L	(DFA6DAT3),A3	;ADDR OF TIMING WORD #3
1086	00000216	2678	0000	0000	E	MOVEA.L	(DFA6OUT3),A4	;ADDR TO OUTPUT TIMING WORD #3
1087	0000021C	3C39	0000	0000	E	MOVE.W	(DFA6MSK3),D6	;MASK USED TO SET TIMING BIT #3
1088	00000222	8D53				OR.W	D6,(A3)	;SET TIMING BIT #3
1089	00000224	3893				MOVE.W	(A3),(A4)	;OUTPUT TIMING WORD #3
1090								
1091								
1092	00000226	4EB9	0000	0000	E	JSR	COARSE	PERFORM 1ST PASS SEARCH
1093								
1094								
1095	0000022C	BD53				EOR.W	D6,(A3)	;CLEAR TIMING BIT #3
1096	0000022E	3893				MOVE.W	(A3),(A4)	;OUTPUT TIMING WORD #3
1097								
1098								
1099	00000230	4A39	0000	0034	R	TST.B	NORES	CHECK STATUS
1100	00000236	6714				BEQ.S	SECSRCH	NO ERRORS, GO TO 2ND SEARCH
1101	00000238	123A	FDFA			MOVE.B	NORES,D1	GET ERROR BIT NUMBER
1102	0000023C	4881				EXT.W	D1	MAKE A WORD
1103	0000023E	302E	0004			MOVE.W	DFR_ERR(A6),D0	GET ERRORS SO FAR
1104	00000242	03C0				BSET	D1,D0	INDICATE ERROR IN 1ST PASS
1105	00000244	3D40	0004			MOVE.W	D0,DFR_ERR(A6)	UPDATE FIELD
1106	00000248	6000	00DE			BRA	SNDRESP	
1107								
1108	0000024C	41ED	0008			SECSRCH	LEA	TFA_FRI(A5),A0
								GET PTR TO FREQ.

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

1109 00000250 303C 0006      MOVE.W  #N10KHZ,D0      ONLY WANT TO 10 KHZ
1110 00000254 4EB9 0000 0000 E JSR    BCDBIN          CONVERT TO BINARY
1111 0000025A 0681 0000 0032     ADD.L  #RNDFRQ,D1      ROUND TO MHZ
1112 00000260 83FC 0064     DIVS   #KHZ,D1       CONVERT TO MHZ
1113 00000264 33C1 0000 003E R   MOVE.W  D1,IFRQMHZ     SAVE FREQ. IN MHZ
1114                                *
1115 0000026A 4EB9 0000 0000 E   JSR    SQTHRESH       DETERMINE QTHRESH
1116                                *
1117 00000270 33EE 0016 0000 R   MOVE.W  DFR_ROLL(A6),ROLL GET ROLL ANGLE
1118                                *
1119                                *
1120 00000278 2679 0000 0000 E   MOVEA.L (DFA6DAT4),A3  :ADDR OF TIMING WORD #4
1121 0000027E 2879 0000 0000 E   MOVEA.L (DFA6OUT4),A4  :ADDR TO OUTPUT TIMING WORD #4
1122 00000284 3C39 0000 0000 E   MOVE.W  (DFA6MSK4),D6  :MASK USED TO SET TIMING BIT #4
1123 0000028A 8D53                OR.W   D6,(A3)         :SET TIMING BIT #4
1124 0000028C 3893                MOVE.W  (A3),(A4)       :OUTPUT TIMING WORD #4
1125                                *
1126                                *
1127 0000028E 4EB9 0000 0000 E   JSR    P2SRCH         PERFORM 2ND PASS SEARCH
1128                                *
1129                                *
1130 00000294 BD53                EOR.W   D6,(A3)       :CLEAR TIMING BIT #4
1131 00000296 3893                MOVE.W  (A3),(A4)       :OUTPUT TIMING WORD #4
1132                                *
1133                                *
1134 00000298 4A39 0000 0054 R   TST.B   NORES2        CHECK STATUS
1135 0000029E 670E 4E71     BEQ    GOODLOB        IF 0 THEN NO ERRORS
1136 000002A2 302E 0004     MOVE.W  DFR_ERR(A6),D0      GET ERRORS SO FAR
1137 000002A6 08C0 0007     BSET    #PFIT,D0      INDICATE ERROR
1138 000002AA 3D40 0004     MOVE.W  D0,DFR_ERR(A6)  UPDATE FIELD
1139                                *
1140 000002AE 3D7A FD9A 0010     GOODLOB  MOVE.W  LOB_DFR_LOB(A6)  RETURN LOB
1141 000002B4 203A FD96     MOVE.L   QUAL,D0      GET QUALITY
1142 000002B8 6A02     BPL.S   FNDMAG        CHECK IF NEGATIVE
1143 000002BA 4480     NEG.L   D0            ALWAYS WANT POSITIVE
1144 000002BC 4241     CLR.W    D1            CLEAR MAGNITUDE COUNT
1145                                *
1146 000002BE 4A80     MAGLOOP  TST.L    D0            CHECK IF 0 YET
1147 000002C0 6706     BEQ.S   RETQUAL        IF 0 THEN DONE
1148 000002C2 E280     ASR.L    #1,D0          ADJUST VALUE
1149 000002C4 5241     ADDQ.W   #1,D1          INCREMENT MAGNITUDE COUNT
1150 000002C6 60F6     BRA.S    MAGLOOP        ENDLOOP
1151                                *
1152 000002C8 3D41 0012     RETQUAL  MOVE.W  D1,DFR_QUAL(A6)  RETURN QUALITY (MAGNITUDE)
1153 000002CC 605A 4E71     BRA     SNDRSP
1154                                *****

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

1155 * * * ARF DF BITE PROCESSING
1156 *
1157 *
1158 *****
1159 ADFBITP CLR.W D4 INITIALIZE RESULTS (ALL FAIL)
1160 LEA BITETBL,A0 GET PTR TO EXPECTED VALUES
1161 CLR.W D0 INITIALIZE INDEX
1162 MOVE.W #NBITE,D1 GET NUMBER OF FREQ.S
1163 MOVE.L INTRQ,D2 GET INTEGER DF FREQUENCY
1164
1165 CMP.L (A0)+,D2 CHECK IF DF FREQ.
1166 BEQ.S BITFRND FOUND ADF BITE FREQ
1167 ADDQ.W #1,D0 BUMP UP INDEX
1168 DBF D1,BITFRLP ENDOLOOP
1169 BRA.S UPDRES FREQ. NOT AN ADF BITE FREQ.
1170
1171 BITFRND ASL.W #5,D0 CONVERT INDEX TO OFFSET
1172 * EACH SET IS 8 LONGWORDS
1173 LEA BITETBL,A2 GET PTR TO FREQ LIST
1174 ADDA.L #BITFRSZ,A2 JUMP OVER FREQ LIST
1175 ADD.W D0,A2 ADD OFFSET FOR FREQ
1176 MOVE.W NBL,D0 GET NUMBER OF BL S
1177 SUBQ.W #1,D0 DECR. FOR COUNTER
1178 MOVE.W D0,D1 GET A COPY OF N-1
1179 LEA DFR_BASE(A6),A1 GET PTR TO COLLECTED VALUES
1180
1181 * LOOP
1182 MOVE.W (A1)+,D2 GET COLLECTED VALUE
1183 SUB.W (A2)+,D2 GET DIFFERENCE WITH EXPECTED
1184 BPL.S TEST IF >=0 THEN OKAY TO TEST
1185 NEG.W D2 GET ABSOLUTE VALUE
1186 CMP.W (A2)+,D2 CHECK WITH DELTA FOR BL
1187 BGT.S FAIL NOT OKAY, BL FAILS
1188 MOVE.W D1,D3 GET A COPY OF N-1
1189 SUB.W D0,D3 GET BIT NUMBER FOR BL
1190 BSET D3,D4 INDICATE BL PASSES
1191 DBF D0,LOOP ENDOLOOP
1192 *
1193 UPDRES MOVE.W D4,DFR_ERR(A6) RETURN BITE RESULTS
1194 CLR.W D6 RESET REPORT FLAG
1195 BRA.S SENDIT SEND DF RESPONSE
1196
1197 * * SEND RESPONSE MESSAGE
1198 *
1199 SDRRESP CLR.W D6 RESET NEED REPORT FLAG
1200 TST.W DFRPEN CHECK IF REPORTING IS ENABLED
1201 BEQ SENDIT NOT ENABLED, DO NOT REPORT
1202 MOVE.B DFR_TYP(A6),D0 GET DF TYPE

```


Improved GUARDRAIL V MC68000 'DF' Files DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

1202	00000338	323C 0002	MOVE.W	#NPETTYPE,D1	GET NUMBER TO CHECK
1203	0000033C	41FA FD1A	LEA	PETTYPE,A0	GET PTR TO PE TYPE TABLE
1204					
1205	00000340	B018	CMP.B	(A0)+,D0	CHECK IF PE TYPE
1206	00000342	670A 4E71	BEQ	REPORT	IF SO WANT TO REPORT
1207	00000346	51C9 FFF8	DBF	D1,PELOOP	ENDLOOP
1208					
1209	0000034A	6020 4E71	BRA	SENDIT	NOT A PE TYPE, NO REPORT
1210					
1211	0000034E	3C3C 0001	MOVE.W	#1,D6	SET NEED REPORT FLAG
1212	00000352	284E	MOVE.L	A6,A4	SAVE DFR PACKET
1213	00000354	4EB9 0000 0000 E	JSR	GETPAK	GET A PACKET FOR DR1
1214					
1215					
1216					
1217					
1218	0000035A	323C 0028	MOVE.W	#DFR,D1	GET NUMBER OF BYTES
1219	0000035E	5341	SUBQ.W	#1,D1	ADJUST FOR INDEX
1220	00000360	204E	MOVE.L	A6,A0	GET A COPY OF UDR1 PTR
1221	00000362	224C	MOVE.L	A4,A1	GET A COPY OF DFR PTR
1222					
1223	00000364	10D9	MOVE.B	(A1)+,(A0)+	FORMAT UDR1 MESSAGE
1224	00000366	51C9 FFFC	DBF	D1,DRILP	ENDLOOP
1225					
1226	0000036A	C94E	EXG	A6,A4	A6 <- DFR, A4 <- UDR1
1227					
1228	0000036C		?SCHED	CPUSPEND,#4,...,A6,A4/A5/D6 ;	SEND DFR MESSAGE
1263					
1264	0000038C	4A46	TST.W	D6	CHECK IF NEED TO REPORT
1265	0000038E	6700 010E	BEQ	DONE	NO REPORT, EXIT
1266					
1267	00000392	197C 004C 0000	MOVE.B	#UDRMID,UDR1_MID(A4)	SET MESSAGE ID.
1268	00000398	322C 0002	MOVE.W	UDR1_ACC(A4),D1	GET ACCOUNTABILITY
1269	0000039C	0041 C000	OR.W	#UNSOLIT,D1	MAKE UNSOLICITED
1270	000003A0	3941 0002	MOVE.W	D1,UDR1_ACC(A4)	
1271					
1272	000003A4		?SCHED	CPUSPEND,#4,...,A4,A5 ;	SEND UDR1 MESSAGE
1307					
1308	000003C4	4EB9 0000 0000 E	JSR	GETPAK	GET A PACKET FOR UDR2 MESSAGE
1309	000003CA	284E	MOVE.L	A6,A4	SAVE UDR2 PACKET PTR
1310	000003CC	4EB9 0000 0000 E	JSR	GETPAK	GET A PACKET FOR UDR3 MESSAGE
1311					
1312	000003D2	197C 004D 0000	MOVE.B	#UDRMID,UDR2_MID(A4)	PUT IN MESSAGE ID.
1313	000003D8	197C 0004 0001	MOVE.B	#DFRNB,UDR2_NIB(A4)	PUT IN BLOCK COUNT
1314	000003DE	322D 0002	MOVE.W	TFA_ACC(A5),D1	GET ACCOUNTABILITY
1315	000003E2	0041 C000	OR.W	#UNSOLIT,D1	MAKE UNSOLICITED
1316	000003E6	3941 0002	MOVE.W	D1,UDR2_ACC(A4)	PUT INTO UDR2

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

1317
1318
1319
1320
1321 000003EA 41F9 0000 0000 E
1322 000003F0 43EC 0004
1323 000003F4 323C 0008
1324
1325 000003F8 22D8
1326 000003FA 51C9 FFFC
1327
1328 000003FE
1329
1330 0000041E 1D7C 004E 0000
1331 00000424 1D7C 0004 0001
1332 0000042A 322D 0002
1333 0000042E 0041 C000
1334 00000432 3D41 0002
1335
1336
1337
1338
1339
1340 00000444 43EE 0014
1341 00000448 41ED 0008
1342 0000044C 323C 0004
1343
1344 00000450 12D8
1345 00000452 51C9 FFFC
1346
1347
1348
1349
1350 00000456 303A FB02
1351 0000045A 907A FFC8
1352 0000045E 3D40 001A
1353
1354 00000462 3D7A FBCE 001C
1355 00000468 3D7A FFC6 001E
1356 0000046E 3D79 0000 0000 E
                                0020
*
* FILL UP UDR2 WITH 9 ELEMENTS FROM SECOND PASS ERROR
* ARRAY (ERROR2)
*
LEA ERROR2,A0 GET PTR TO ERROR2 ARRAY
LEA UDR2,ER2(A4),A1 GET PTR TO SPACE IN UDR2
MOVE.W #NDRZER2,D1 GET NUMBER OF ITEMS
*
DR2LP MOVE.L (A0)+,(A1)+ FILL IN UDR2
DBF D1,DR2LP ENDLOOP
*
?SCHED CPUSEND,#4,,,A4,A5/A6/A0 ; SEND UDR2
*
MOVE.B #UDR3MID,UDR3_MID(A6) PUT IN MESSAGE ID.
MOVE.B #DFRNB,UDR3_NIB(A6) PUT IN BLOCK COUNT
MOVE.W TFA_ACC(A5),D1 GET ACCOUNTABILITY
OR.W #UNSOLIT,D1 MAKE UNSOLICITED
MOVE.W D1,UDR3_ACC(A6) PUT INTO UDR3
*
* FILL UP UDR3 WITH REMAINING ERROR2 ELEMENTS
*
LEA UDR3,ER2(A6),A1 GET PTR TO PACKET SPACE
MOVE.W #NDRZER2,D1 GET NUMBER OF ELEMENTS
*
DR3LP MOVE.L (A0)+,(A1)+ FILL IN UDR3
DBF D1,DR3LP ENDLOOP
*
* PUT FREQ. INTO UDR3
*
LEA UDR3,FRI(A6),A1 GET PTR TO UDR3 FREQ.
LEA TFA_FRI(A5),A0 GET PTR TO DF FREQ.
MOVE.W #NFREQB,D1 GET NUMBER OF FREQ BYTES
*
DR3LP2 MOVE.B (A0)+,(A1)+ GET DF FREQ.
DBF D1,DR3LP2
*
* ADJUST BADBL TO BE A NUMBER FROM 1 ON
* (NOT 8 DOWN)
*
MOVE.W NBL,D0 GET NUMBER OF BASELINES
SUB.W BADBL,D0 REVERSE BADBL NUMBER
MOVE.W D0,UDR3_BAD(A6) GET BAD BL NO.
*
MOVE.W MINX1,UDR3_MN1(A6) GET MIN FROM SEARCH 1
MOVE.W MINVAL1,UDR3_MV1(A6) GET VALUE AT MINX1
MOVE.W SCHIN,UDR3_SCH(A6) GET SECOND MINIMUM

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

```

1397 00000476 3D79 0000 0000 E      MOVE.W  SCVAL,UDR3_SCV(A6) GET VALUE AT SEC. MIN.
      0022
1398
1399 0000047E      *      ?SCHED CPUSEND,#4,...A6,A5 ; SEND UDR3
1434      *
1435      *
1436 0000049E 2C4D      DONE      MOVE.L  A5,A6      GET INPUT PACKET
1437 000004A0 4EB9 0000 0000 E      JSR      PUTLPK      RETURN PACKET
1438      *
1439      *
1440 000004A6 2679 0000 0000 E      MOVEA.L (DFA6DAT1),A3 ; ADDR OF TIMING WORD #1
1441 000004AC 2879 0000 0000 E      MOVEA.L (DFA6OUT1),A4 ; ADDR TO OUTPUT TIMING WORD #1
1442 000004B2 3C39 0000 0000 E      MOVE.W  (DFA6MSK1),D6 ; MASK USED TO CLEAR TIMING BIT #1
1443 000004B8 BD53      EOR.W   D6,(A3) ; CLEAR TIMING BIT #1
1444 000004BA 3893      MOVE.W  (A3),(A4) ; OUTPUT TIMING WORD #1
1445      *
      ?EXIT
1446 000004BC      END
1460 000004C6

```

ASSEMBLER ERRORS - 0

SYMBOL TABLE

ABORT	00000002	ACCMID	00000000	ACCRMID	00000001
ADFBIT	R 00000000	ADFBITP	R 000002D0	AINTMID	00000002
ALORMID	00000003	ARFINMID	00000004	ASTAMID	00000005
ASTRMID	00000006	BADBL	R 00000024	BANDLB	R 00000040
BANGLMID	00000007	BCMBIN	E 00000016	BDFMID	0000000E
BITEMID	00000009	BITETBL	E 0000001C	BITMID	00000008
BITQMID	0000000A	BLSGA	00000004	BLSGF	00000003
BLSGP	R 0000014A	BLSGPRS	E 00000013	BTFRFND	R 000002EE
BTFRLP	R 000002E2	BTFRSZ	00000018	BTMID	0000000B
BTQMID	0000000C	BUFFER	R 00000002	CALCMID	0000000D
COARSE	E 00000017	CPUBMID	0000000F	CFURMID	00000010
CPUSEND	E 0000001A	CTLMID	00000011	DELTA	R 00000056
DFA6DAT1	E 00000005	DFA6DAT2	E 00000008	DFA6DAT3	E 0000000B
DFA6DAT4	E 0000000E	DFA6E	R 00000060	DFA6MSK1	E 00000007
DFA6MSK2	E 0000000A	DFA6MSK3	E 0000000D	DFA6MSK4	E 00000010
DFA6OUT1	E 00000006	DFA6OUT2	E 00000009	DFA6OUT3	E 0000000C
DFA6OUT4	E 0000000F	DFFMID	00000013	DFMID	00000012
DFFRMID	00000014	DFR	00000028	DFREJ	00000000
DFRJMID	00000016	DFRMID	00000015	DFRNIB	00000004
DFRPN	E 00000011	DFR_ACC	00000002	DFR_ARF	00000304
DFR_BASE	00000018	DFR_BLK	00000300	DFR_ERR	00000004
DFR_HEA	00000014	DFR_LAT	00000008	DFR_LOB	00000010
DFR_LON	0000000C	DFR_MID	00000000	DFR_NIB	00000001

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

DFR_QUAL	00000012	DFR_ROLL	00000016	DFR_SSM	00000007
DFR_TYP	00000006	DCMID	00000017	DIREV	E 00000015
DONE	R 0000049E	DFMID	00000018	DR1LP	R 00000364
DR2LP	R 000003F8	DR3LP	R 0000043E	DR3LP2	R 00000450
DSACMID	00000019	DSACMID	0000001A	DSACMID	0000001B
DSEMD	0000001C	DSEMD	0000001D	DSEMD	0000001E
DSPRMID	0000001F	DSRMID	00000020	ERRMSK	00000024
ERROR2	E 00000002	FAIL	R 0000031C	FATAL1	00000004
FINENAV	R 0000017A	FINMAG	R 000002BC	FRHPR	R 0000003A
FLRPTR	R 00000036	GETPAK	E 00000012	GINTMID	00000021
GLOBMID	00000022	GOODLOB	R 000002AE	GPROJMID	00000023
GSACMID	00000025	GSACMID	00000024	GSACMID	00000026
GSAOIMID	00000028	GSDAMID	00000027	GSEMD	00000029
GSMID	0000002A	GSPRMID	0000002B	GSRBTMID	0000002C
GSMID	0000002D	GSPRMID	0000002E	GTBADBL	R 0000013A
GTNUM	R 000001AE	IFRQHZ	R 0000003E	INQUAD	E 00000014
INTFRQ	R 00000026	ISECTMID	R 0000002F	KHZ	00000064
LLLOBMID	00000030	LOB	R 0000004A	LOOP	R 0000030A
MAGLOOP	R 0000028E	MAXROLL	E 00000001	MIDMODE	00000005
MIDPRI	00000007	MIDROUTE	00000000	MIDSIZE	00000006
MIDTYP	00000004	MINVAL1	R 00000030	MINVAL2	R 00000044
MINX1	R 00000032	MINX2	R 00000048	MISSEG	00000008
MLREQMID	00000031	MODLCMID	00000032	MULT	00000006
N10KHZ	00000006	NAIDBIT	R 000000EA	NARG	00000000
NAV	0000000C	NAVDATA	E 00000000	NAV_HEAD	00000008
NAV_LAT	00000000	NAV_LONG	00000004	NAV_ROLL	0000000A
NBITE	00000005	NBL	R 0000002A	NDR2ER2	00000008
NDR3ER2	00000003	NREQ	0000000A	NREQB	00000004
NOCAL	00000009	NORES	R 00000034	NORES2	R 00000054
NOSIGNL	00000002	NPEYPE	00000002	NTAB	R 0000005B
PZSRCH	E 00000018	PELOOP	R 00000340	PETYPE	R 00000058
PFIT	00000007	POSROLL	R 0000019A	PULPAK	E 0000001B
QTHRESH	R 00000050	QUAL	R 0000004C	RCB	00000000
RCBA0	00000026	RCBA1	0000002A	RCBA2	0000002E
RCBA3	00000032	RCBA4	00000036	RCBA5	0000003A
RCBA6	0000003E	RCBA7	0000005C	RCBA8	00000060
RCBA9	00000064	RCBA0	00000006	RCBDMK	00000058
RCBLEN	00000067	RCBLNK	00000000	RCBMFG	0000004C
RCBNUL	00000054	RCBPC	00000046	RCBPRTY	0000004E
RCBREGS	00000006	RCBSP	00000042	RCBSR	0000004A
RCBSTAT	00000052	RCBTIB	00000002	RCBWAIT	00000004
REPORT	R 0000034E	REQUAL	R 000002C8	RNDFRQ	00000032
ROLL	R 00000042	ROLLANG	00000005	SICMID	00000038
SIC2MID	00000039	SICMID	00000037	SIMMID	00000033
SIOIMID	0000003A	SIO2MID	0000003B	SITNEMID	0000003C
SZC1MID	0000003E	SZC2MID	0000003D	SZMMID	00000034
SZC2MID	00000040	SZC2MID	00000041	SZTNEID	00000042

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.LST

SRBMID	0000003F	SC0DSMID	00000035	SC0GSMID	00000036
SCANLMID	00000043	SCCALMID	00000044	SCMIN	E 00000003
SCROMID	00000045	SCVAL	E 00000004	SCVTMID	00000046
SDMID	00000047	SECSRCH	R 0000024C	SENDIT	R 0000036C
SIGPRS	R 00000104	SNDRSP	R 00000328	SQTHRESH	E 00000019
STAMID	00000048	TAOI	00000007	TAUTO	00000004
TBITE	00000001	TCALIB	00000006	TDATAMID	00000049
TEST	R 00000312	TFA	000000A2	TFAMID	0000004B
TFA_ACC	00000002	TFA_ASUB	00000007	TFA_AUD	00000007
TFA_BAEI	0000000E	TFA_BAEQ	00000092	TFA_BAND	0000000D
TFA_BASE	0000000E	TFA_CONT	00000004	TFA_ERR	00000004
TFA_FINE	00000006	TFA_FR1	00000008	TFA_FR2	00000009
TFA_FR3	0000000A	TFA_FR4	0000000B	TFA_FR5	0000000C
TFA_HEA	0000009E	TFA_LAT	00000096	TFA_LON	0000009A
TFA_MID	00000000	TFI_RESP	00000005	TFA_ROLL	000000A0
TFA_SEG	00000300	TFA_SSM	00000006	TFA_TYP	00000001
TGEO	00000008	TGS	00000005	THETA	R 0000002C
TINTEROP	00000003	TLINKMID	0000004A	TMANUAL	00000002
TOOLNG	00000001	TOTMID	0000004F	TRUE	00000001
TSTROLL	R 00000192	UDR1	00000028	UDR1MID	0000004C
UDR1_ACC	00000002	UDR1_ARF	00000304	UDR1_BAS	00000018
UDR1_BLK	00000300	UDR1_ERR	00000004	UDR1_HEA	00000014
UDR1_LAT	00000008	UDR1_LOB	00000010	UDR1_LON	0000000C
UDR1_MID	00000000	UDR1_NIB	00000001	UDR1_QUA	00000012
UDR1_ROL	00000016	UDR1_SSM	00000007	UDR1_TYP	00000006
UDR2	00000028	UDR2MID	0000004D	UDR2_ACC	00000002
UDR2_ARF	00000304	UDR2_BLK	00000300	UDR2_ER2	00000004
UDR2_MID	00000000	UDR2_NIB	00000001	UDR3	00000024
UDR3MID	0000004E	UDR3_ACC	00000002	UT-3 ARF	00000304
UDR3_BAD	0000001A	UDR3_BLK	00000300	UDR3_ER2	00000004
UDR3_FR1	00000014	UDR3_FR2	00000015	UDR3_FR3	00000016
UDR3_FR4	00000017	UDR3_FR5	00000018	UDR3_MID	00000000
UDR3_MW1	0000001C	UDR3_MW1	0000001E	UDR3_NIB	00000001
UDR3_SCM	00000020	UDR3_SCV	00000022	UNSLIT	0000C000
UPPRES	R 00000320				

Improved GUARDRAIL V MC68000 'DF' Files

18 FILES WITH EXTENSION '.PIC'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DF.PIC_NUM

18.1 DF.PIC_NUM

```

***** Source Listing ==> DF.PIC_NUM *****
*****
1 .NOFILL
2 .LM -7
3
4     DF      DF REQUEST (GDPU -> ADPU)
5
6     All DF requests to the ARFs.
7
8     1) DF_FR1 - DF_FR5  FREQUENCY OF DF
9       Frequency consists of 10 BCD digits. The most
10      significant digit represents 1 GHZ.
11
12     2) DF_GAIN  RECEIVER GAIN SETTINGS
13       This field is ignored by ADPU software.
14
15     3) DF_BAND  BANDWIDTH VALUES
16       0 = 8 KHZ
17       1 = 15 KHZ
18       2 = 50 KHZ
19
20     4) DF_DET   DETECTOR VALUES
21       0 = FM
22       2 = AM
23       4 = CW
24       5 = LSB
25       6 = USB
26       7 = ISB
27
28     5) DF_TC   AGC TIME CONSTANT
29       This field is ignored by ADPU software
30
31     6) DF_DU   AGC DUMP CONSTANT
32       This field is ignored by ADPU software.
33
34     7) DF_SQ   SQUELCH
35       This field is ignored by ADPU software.
36
37     8) DF_CV   COARSE VERIFY FLAG
38       This flag is used only during calibration verify.
39       0 = fine DF
40       1 = coarse DF
41
42     9) DF_POL  ANTENNA POLARIZATION
43       This field is ignored by ADPU software.

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DF.PIC NUM

```

44 10) DF_TYP  DF TYPE DESIGNATOR
45     1 = BITE
46     2 = MANUAL DF
47     3 = INTEROP
48     4 = AUTO DF (DS)
49     5 = GENERAL SEARCH (FINE)
50     6 = CALIBRATION
51     7 = AOI
52     8 = GEOSCREEN
53
54
55 11) DF_AUDIO  AUDIO CORRELATOR RECEIVER TUNING
56     0 = do not tune audio correlator receiver
57     1 = tune audio correlator receiver
58
59 12) DF_FINE  COARSE/FINE DF FLAG
60     0 = coarse DF
61     1 = fine DF
62
63 13) DF_RESP  RESPONSE FLAG
64     0 = no response
65     1 = send response
66
67 14) DF_CONT  CONTIGUOUS FLAG
68     0 = non contiguous DF
69     1 = contiguous segments---ie. do no re-tune receiver
70
71 15) DF_SEG  DF SEGMENT NUMBER
72     Valid segment numbers are from 1 thru 6.
73
74 16) DF_EN1_DF_EN2_ENABLE ARFX REPORTING
75     0 = Do not send additional DF data messages (UDR1,UDR2,UDR3)
76     1 = Do send additional DF data messages.
77
78
79 BYTES DESCRIPTION      BYTE 0  BYTE 1  BYTE 2  BYTE 3  BYTES
80 -----
81 1 MESSAGE ID ($12)      DF_MID
82 1 ARF ID. & BLOCK COUNT  DF_NIB
83 4 ARF ID.              **** DF_ARF
84 4 BLOCK COUNT          **** DF_BLK
85 2 ACCOUNTABILITY       DF_ACC
86
87 FREQUENCY SEE NOTE 1
88 1 1ST 2 BCD DIGITS     DF_FRQ
89 1 2ND 2 BCD DIGITS     DF_FR1
90 1 3RD 2 BCD DIGITS     DF_FR2  DF_FR3

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DF.PIC_NUM

91	1	4TH 2 BCD DIGITS	7
92			
93	1	LAST 2 BCD DIGITS	8
94	1	GAIN TYPE & MANUAL CNTR	9
95	2	GAIN TYPE	
96	6	MANUAL GAIN SETTING	
97		RECEIVER R-924 CONTROL	
98	1	BANDWIDTH AND DETECTORS	10
99	3	BANDWIDTH CODE SEE NOTE	10
100	3	DETECTOR CODE SEE NOTE	
101	1	AGC TIME CONSTANT	
102	1	AGC DUMP CONSTANT	
103	1	MISC FLAGS	
104	1	SQUELCH	11
105	1		
106	1	ENABLE ARF1 REPORTING	
107	1	ENABLE ARF2 REPORTING	
108	1	COARSE VERIFY FLAG	
109	2	ANTENNA POLARIZATION	
110			
111	1	DF TYPE SEE NOTE 4	
112	1	DF SUB TYPE & AUDIO COR	12
113	1	AUDIO CORR RCVR TUNE	13
114	1	COARSE/FINE FLAG	
115	1	RESPOND FLAG	
116	1	CONTIGUOUS FLAG	
117	4	SEGMENT NUMBER	
118			
119			
120		TOTAL IS 14 BYTES FOR DF	
121		LM R	
122		.FILL	

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DF1.PIC_NUM

18.2 DF1.PIC_NUM

***** Source Listing ==> DF1.PIC_NUM *****

1 .NOFILL

2 .LM -7

3

4

DF1 DF REQUEST QUEUE ITEM

5

THERE WILL BE ONE QUEUE FOR EACH OF THE TYPES OF FINE
 DF AND CALIBRATION, AND ONE QUEUE FOR THE COARSE DF
 REQUESTS. THE DIFFERENT QUEUES AND THE CORRESPONDING
 QUEUE DEPTH ARE SHOWN IN THE TABLE. THE QUEUES WILL BE
 IMPLEMENTED AS A SET SIZE (THE DEPTH) OF POINTERS TO
 THE QUEUE ITEMS. THE QUEUES WILL BE IN LOCAL COMMON, AND
 POINTERS TO THE TOP AND BOTTOM OF EACH QUEUE WILL BE
 MAINTAINED.

QUEUE DEPTH

BITE 9

MANUAL 14

INTEROP. 7

AUTO DF 14

G.S. 4

CALIP. 5

COARSE 18

BYTES DESCRIPTION

2 TIME ENTERED QUEUE

1 MESSAGE ID

1 UNUSED

2 ACCOUNTABILITY

1 DF TYPE

1 DF SUB TYPE & AUDIO COR

1 AUDIO CORRELATION TUNE

7 DF SUB TYPE

FREQUENCY (10 BCD DIGIT

1 1ST 2 BCD DIGITS

1 2ND 2 BCD DIGITS

1 3RD 2 BCD DIGITS

1 4TH 2 BCD DIGITS

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTES
DF1_TIM	DF1_MID	UNUSED		0- 1
				2
				3
DF1_ACC	DF1_TYP	DF1_ASUB		4- 5
	DF1_AUD	DF1_SUB	*****	6
				7
DF1_FR1	DF1_FR2	DF1_FR3	DF1_FR4	8
				9
				10
				11

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DF1.PIC_NUM

44	1	LAST 2 BCD DIGITS	DF1_FR5	12
45		RECEIVER R 324 CONTROL	DF1_RCVR	13
46	1	GAIN TYPE & MANUAL CNTR	DF1_GAIN	13
47	2	GAIN TYPE	** DF1_TYPE	
48	6	MANUAL GAIN SETTING	***** DF1_MAN	
49	1	BANDWIDTH AND DETECTORS	DF1_RNG	14
50	1	MISC FLAGS	DF1_FLAG	15
51				
52				
53	1	DS PORTION NO.	DF1_FOR	16
54	1	DS ENTRY NO.	DF1_DSE	17
55	4	GS SIGNAL TYPE BIT MAP	DF1_SIG	18- 21
56				
57	1	ARF ID	DF1_ARF	22
58	1	DS BANDWIDTH	DF1_BND	23
59				
60	1	AUTO DF COUNTER	DF1_ADFC	24
61	1	ART FLAG	DF1_ARTF	25
62				
63				
64		TOTAL IS 26 BYTES FOR DF1		
65		LM R		
66		FILL		

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFC.PIC_NUM

18 3 DFC.PIC_NUM

***** Source Listing -> DFC.PIC_NUM *****

1 .NOFILL
 2 .LM -7

3
 4 DFC DF DATA COLLECTION SEQUENCE HEADER.

5
 6 EACH HEADER BLOCK CONTAINS THE POINTERS TO
 7 THE APPROPRIATE COMMAND SEQUENCE FOR THE
 8 RF PROCESSOR, ACU'S, AND FAST DF CONTROLLER.
 9

10	BYTES DESCRIPTION	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTES
11						
12						
13	2 NUMBER OF STEPS IN SEQU	DFC_NUM				0- 1
14	4 PTR TO RF PROCESSOR COM			DFC_RFP		2- 5
15						
16	4 PTR TO ACU COMMANDS			DFC_ACU		6- 9
17						
18	4 PTR TO FDRC COMMANDS			DFC_FDRC		10- 13
19						

20
 21 TOTAL IS 14 BYTES FOR DFC
 22 .LM R
 23 .FILL

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFF.PIC_NUM

1A 4 DFF.PIC_NUM

***** Source Listing -> DFF.PIC_NUM *****

```

1  NOFILL
2  LM 7
3
4      DFF      OF FREQUENCY FOR SIGNAL GENERATOR
5
6      1) FREQUENCY CONSISTS OF 10 BCD DIGITS. THE MOST
7        SIGNIFICANT DIGIT REPRESENTS 1 GHE.
8
9      2) BANDWIDTH VALUES
10
11         0 - 8 KHZ
12         1 - 15 KHZ
13         2 - 50 KHZ
14
15      3) DETECTOR VALUES
16
17         0 - FM
18         2 - AM
19         4 - CW
20         5 - LSB
21         6 - USB
22         7 - ISR
23
24

```

BYTES DESCRIPTION	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTES
27 1 MESSAGE ID. (\$13)	DFF_MID				0
28 1 1ST 2 BCD DIGITS SEE NO		DFF_FR1			1
29 1 2ND 2 BCD DIGITS			DFF_FR2		2
30 1 3RD 2 BCD DIGITS				DFF_FR3	3
31					
32 1 4TH 2 BCD DIGITS	DFF_FR4				4
33 1 LAST 2 BCD DIGITS		DFF_FR5			5
34 1 BANDWIDTH & DETECTOR			DFF_RNG		6
35 3 BANDWIDTH CODE SEE NOTE		DFF_BAND ***			
36 3 DETECTOR CODE SEE NOTE			DFF_DET ***		
37					
38					
39 TOTAL IS 7 BYTES FOR DFF					
40 LM R					
41 FILL					

Improved GUARDRAIL V MC68000 'DF' Files DRCL: [ALGO.IGR.MC68000_DF]DFPR.PIC_NUM

1R 5 DFPR.PIC_NUM

***** Source [listing] --> DFPR.PIC_NUM *****

```

1  NOFILL
2  LM 7
3
4  DFPR  DF RESPONSE (GDPV -> PE)
5
6  MESSAGE TO TRANSFER DF RESPONSE INFORMATION
7  (TWO LOOPS)
8
9  1) FREQUENCY CONSISTS OF 10 BCD DIGITS. THE MOST
10 SIGNIFICANT DIGIT REPRESENTS 1 GHZ.
11
12 2) DF ERRORS BIT MAP BIT ASSIGNMENTS
13
14 0 DF REJECTED (PREV. TOO LONG )
15 1 DF DID NOT COMPLETE IN SEGMENT
16 2 NO SIGNAL PRESENT
17 3 BASE LINE SIG PRES FAILED
18 4 BASE LINE SIG PRES ABORT
19 5 ROLL ANGLE EXCEEDS LIMIT
20 6 MULTIPLE MINIMA IN 1ST SEARCH
21 7 POOR FIT, BAD QUALITY
22 8 MISSING SEGMENT (FINE DF)
23 9 NO CAL TABLES
24
25 3) DF TYPE VALUES
26
27 1 = BITE
28 2 = MANUAL DF
29 3 = INTEROP
30 4 = AUTO DF (DS)
31 5 = GENERAL SEARCH (FINE)
32 6 = CALIBRATION
33 7 = AOI
34 8 = GEOSCREEN
35
36
37 BYTES DESCRIPTION      BYTE 0      BYTE 1      BYTE 2      BYTE 3      BYTES
38 -----
39 1 MESSAGE ID (S14)      DFPR_MID
40 1 DF REJECTED FLAG      DFPR_REJ
41 2 ACCOUNTABILITY      DFPR_ACC
42
43 1 AUDIO CORRELATION FLAG DFPR_ACR

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFPR.PIC.NUM

44	1	COUNTER FOR DS	DFPR_CO	5
45	1	PORTION NUMBER (1-14)	DFPR_POR	6
46	1	ENTRY NO FOR DS (1-70)	DFPR_ENT	7
47				
48	4	SIGNAL TYPE BIT MAP	DFPR_SIG	8- 11
49				
50		DF FREQUENCY SEE NOTE	DFPR_FRQ	12
51	1	MSB OF FREQ (BCD)	DFPR_FR1	12
52	1	2ND BYTE OF FREQ	DFPR_FR2	13
53	1	3RD BYTE OF FREQ	DFPR_FR3	14
54	1	4TH BYTE OF FREQ	DFPR_FR4	15
55				
56	1	LSB OF FREQ	DFPR_FR5	16
57		FIRST LOP	DFPR_LP1	17
58	1	ARF ONE ID	DFPR_AR1	17
59	2	ERROR CODE SEE NOTE 2	DFPR_ER1	18- 19
60				
61	1	DF TYPE SEE NOTE 3	DFPR_TY1	20
62	1	SIGNAL STRENGTH MEASURE	DFPR_SS1	21
63	2	PAD FOR ALIGNMENT	UNUSED	22- 23
64				
65	4	LATITUDE (BAM)	DFPR_LA1	24- 27
66				
67	4	LONGITUDE (BAM)	DFPR_LO1	28- 31
68				
69	2	LINE OF BEARING (BAM)	DFPR_LB1	32- 33
70	2	QUALITY	DFPR_QU1	34- 35
71				
72	2	HEADING (BAM)	DFPR_HE1	36- 37
73	2	AIRCRAFT ROLL (BAM)	DFPR_RO1	38- 39
74				
75	16	5.6 OR 8 BASELINE PHASE	DFPR_BA1	40- 55
76				
77				
78				
79				
80	2	PAD FOR ALIGNMENT	UNUSED	56- 57
81		SECOND LOP	DFPR_LP2	58
82	1	ARF TWO ID	DFPR_AR2	58
83	1	UNUSED	UNUSED	59
84				
85	2	ERROR CODE SEE NOTE 2	DFPR_ER2	60- 61
86	1	DF TYPE SEE NOTE 3	DFPR_TY2	62
87	1	SIGNAL STRENGTH MEASURE	DFPR_SS2	63
88				
89	4	LATITUDE (BAM)	DFPR_LA2	64- 67
90				

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFPR.PIC_NUM

91	4	LONGITUDE	(RAM)	DFPR_LO2	68- 71
92					
93	2	LINE OF BEARING	(RAM)	DFPR_LB2	72- 73
94	2	QUALITY		DFPR_QU2	74- 75
95					
96	2	HEADING	(RAM)	DFPR_HE2	76- 77
97	2	AIRCRAFT ROLL	(RAM)	DFPR_RO2	78- 79
98					
99	16	5.6 OR 8 BASELINE PHASE		DFPR_BA2	80- 95
100					
101					
102					
103					
104					

TOTAL IS 96 BYTES FOR DFPR
 LM R
 107 FILL

18.7 DFR.PIC_NUM

***** Source Listing ==> DFR.PIC_NUM *****

```

1 .NOFILL
2 .LM -7
3
4 DFR DF RESPONSE (ADPU -> GDPU)
5
6 MESSAGE TO TRANSFER DF RESPONSE INFORMATION
7
8 1) DFR_ERR DF ERRORS BIT MAP BIT ASSIGNMENTS
9 0 - DF REJECTED (PREV. TOO LONG )
10 1 - DF DID NOT COMPLETE IN SEGMENT
11 2 - NO SIGNAL PRESENT
12 3 - BASE LINE SIG PRES FAILED
13 4 - BASE LINE SIG PRES ABORT
14 5 - ROLL ANGLE EXCEEDS LIMIT
15 6 - MULTIPLE MINIMA IN 1ST SEARCH
16 7 - POOR FIT, BAD QUALITY
17 8 - MISSING SEGMENT (FINE DF)
18 9 - NO CAL TABLES
19
20 2) DFR_TYP DF TYPE VALUES
21
22 1 - BITE
23 2 - MANUAL DF
24 3 - INTEROP
25 4 - AUTO DF (DS)
26 5 - GENERAL SEARCH (FINE)
27 6 - CALIBRATION
28 7 - AOI
29 8 - GEOSCREEN
30
31 3) DFR_SSM SIGNAL STRENGTH MEASURE
32 This field contains the digitized AGC voltage which is a
33 crude measure of signal strength. Division of this field
34 by 1.5 should give and approximation of SNR in dB. This
35 field is only valid for calibration measurements.
36
37 4) DFR_LAT LATITUDE
38 Latitude of ARF at time of measurement in BAM.
39
40 5) DFR_LON LONGITUDE
41 Longitude of ARF at time of measurement in BAM.
42
43 6) DFR_LOB LINE OF BEARING

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFR.PIC_NUM

Line of bearing to emitter with respect to ARF heading.

7) DFR_QUAL QUALITY OF CAL TABLE FIT
 Quality of the base line measurements cal table fit. Valid values are 0 thru 32 with 0 being a perfect fit to the cal table.

8) DFR_HEA HEADING
 Heading of ARF at time of measurement

9) DFR_ROLL ROLL ANGLE
 Roll angle of ARF at time of measurement

10) DFR_BASE BASELINE PHASE MEASUREMENTS
 Baseline phase measurements: 5 values for VHF LO, 8 values for VHF HI and 6 values for UHF.

BYTES DESCRIPTION	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTES
1 MESSAGE ID (\$15)	DFR_MID	DFR_NIB			0
1 ARF ID & BLOCK COUNT		**** DFR_ARF			1
4 ARF ID		**** DFR_BLK			2- 3
4 BLOCK COUNT			DFR_ACC		4- 5
2 ACCOUNTABILITY					6
2 ERROR CODE SEE NOTE 1	DFR_ERR	DFR_TYP		DFR_SSM	7
1 DF TYPE SEE NOTE 2					8- 11
1 SIGNAL STRENGTH MEASURE		DFR_LAT			12- 15
4 LATITUDE (BAM)			DFR_LON		16- 17
4 LONGITUDE (BAM)				DFR_QUAL	18- 19
2 LINE OF BEARING (BAM)	DFR_LOB				20- 21
2 QUALITY					22- 23
2 HEADING (BAM)	DFR_HEA			DFR_ROLL	24- 39
2 AIRCRAFT ROLL (BAM)					
16 5,6 OR 8 BASELINE PHASE			DFR_BASE		

TOTAL IS 40 BYTES FOR DFR

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFR.PIC_NUM

91 .LM R
92 .FILL

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFRJ.PIC_NUM

18.8 DFRJ.PIC_NUM

***** Source Listing ==> DFRJ.PIC_NUM *****

```

1 .NOFILL
2 .LM -7
3
4 DFRJ DF REJECTED MESSAGE
5
6 1) DF TYPE VALUES
7
8 1 = BITE
9 2 = MANUAL DF
10 3 = INTEROP
11 4 = AUTO DF (DS)
12 5 = GENERAL SEARCH (FINE)
13 6 = CALIBRATION
14 7 = AOI
15 8 = GEOSCREEN
16
17

```

BYTES DESCRIPTION	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTES
1 MESSAGE ID. (\$16)	DFRJ_MID				0
1 MESS. ID. OF REQUEST		DFRJ_MDR			1
2 ACCOUNTABILITY OF REQUE			DFRJ_ACC		2- 3
1 DF TYPE SEE NOTE 1	DFRJ_TYP				4
1 AUDIO CORR. & SUBTYPE		DFRJ_ASU			5
1 AUDIO CORRELATION		* DFRJ_AUD			
7 SUBTYPE		*****	DFRJ_SUB		
TOTAL IS 6 BYTES FOR DFRJ					
.LM R					
.FILL					

Improved GUARDRAIL V MC68000 'DF' Files

19 FILES WITH EXTENSION '.S68'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA1.S68_NUM

19.1 DFA1.S68_NUM

```
***** Source Listing ==> DFA1.S68_NUM *****
*****
1  TTL      'DFA1'
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *$INCLUDE DFA1.FMT/G
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *

VERSION  DATE
1.0      6/16/83
1.1      5/23/84
1.2      8/20/84  CHECK FOR REPORT ENABLE

I/O DEVICES USED: NONE

ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
ADMINUC

METHOD:

PDL:

?DLINK IPACK      /* GET DF REQUEST */

/* IPACK POINTS TO A PACKET CONTAINING DF REQUEST */

?SCHED DFA4.#3.#1.#1 /* TIMEOUT FOR NAV DATA INPUT */
IF ARFID = ARF1
THEN /* CHECK FLAG FOR ARF1 */
IF ARF1 ENABLED /* CHECK FLAG IN DF MESSAGE */
THEN
DFRPN <- TRUE /* SET FLAG */
ELSE
DFRPN <- FALSE /* ELSE DISABLE */
ENDIF
ELSE /* THIS IS ARF2 */
IF ARF2 ENABLED
THEN
DFRPN <- TRUE
ELSE
DFRPN <- FALSE
ENDIF
ENDIF
IF TUNE AUDIO CORR RCVR /* CHECK FLAG IN REQUEST */
THEN /* TIME IT */
FORMAT TUNING STRING
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA1.S68_NUM

```

44 *      ?WRITW DSOB,#N,BUFAD /* SEND STRING TO RCVR */
45 *      SET TO AGC ATTACK
46 *      ?WRITW DSOB,#N,BUFAD /* TUNE THE RCVR */
47 *      ENDIF
48 *      ?SCHED DFA2,#2,..IPACK /* SCHED DFA2, START DATA COLL */
49 *
50 *      ?EXIT
51 *
52 *
53 *
54 *
55 DFA1  IONT      -M
56      OPT
57 *
58      XDEF      DFA1ST
59      XREF      DFA4
60      XREF      AGSEND
61      XREF      DFA2
62      XREF      DSOB
63      XREF      GETPAK
64      XREF      PUTPAK
65      XREF      FMIFRQ
66      XREF      DFRPEN
67      XREF      ARFID
68 *
69 *SINCLUDE STRC.MAC/S
70 *
71 *SINCLUDE DF.PAC/G
72 *
73 *SINCLUDE DFR.PAC/G
74 *
75 *SINCLUDE ARTEMACS.S/G NOLIST,?DLINK,?EXIT,?SCHED,?WRITW
76 *
77 *SINCLUDE ARTEDATA.S/G NOLIST,RCB
78 *
79      NOLIST
80 *SINCLUDE WIDEQU.EQU/G
81      LIST
82 *SINCLUDE DFERRS.EQU
83 *
84 *
85      SECT      ADMINUC
86 *
87 IPACK  DS.L      1
88 PACHEAD EQU      4
89 DFRNIB EQU      204
90 *

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO, IGR, MC68000_DF] DFA1.S68_NUM

91	BUFAD	DC.L	PARMBLK	POINTER TO PARAM. BLOCK
92	PARMBLK	DC.L	FMTLST	
93		DC.L	TSTRING	
94	*			
95	N	EQU	6	NO. OF ELEMENTS TO SEND
96	*			
97	TSTRING	DS.W	1	1ST 4 BCD DIGITS
98		DS.W	1	2ND 4 BCD DIGITS
99		DC.W	\$C0	AGC SELECT
100	RNG	DS.W	1	BAND WIDTH, DET, AGC CON, AGC DUMP
101	GHZ	DS.W	1	GHZ BIT
102		DC.W	0	
103	*			
104	FMTLST	DC.W	\$0010	1ST WORD OF FREQ
105		DC.W	\$0010	2ND WORD OF FREQ.
106		DC.W	\$0008	AGC / MGC
107		DC.W	\$0008	DET., BW, AGC TC, AGC DUMP
108		DC.W	\$0006	GHZ BIT
109		DC.W	\$0002	
110	*			
111	AGCTC	EQU	1	AGC TIME CONSTANT BIT
112	AGCDUMP	EQU	0	AGC DUMP/ATTACK BIT
113	COUNT	EQU	285	
114	GHZFID	EQU	\$F0	GHZ FIELD MASK
115	ENABLE	EQU	1	ENABLE FLAG
116	ARF1	EQU	1	ARF1 ID VALUE
117	*			
118	DFA1ST	?DLINK	IPACK	; GET PACKET POINTER
119		?SCHED	DFA4, #5, #1, #1	
120		MOVE.L	IPACK, A2	GET PACKET POINTER
121		CLR.L	D2	CLEAR OUT INITIAL BIT VALUE
122		MOVE.W	#DF_EN1, D2	ASSUME ARF1
123		MOVE.B	DF_FLAGS(A2), D0	GET OVERALL FIELD
124		CMF.B	#ARF1, ARFID	CHECK IF ARF1
125		BEQ.S	CHECK	IT IS SO GO CHECK
126		MOVE.W	#DF_EN2, D2	SET BIT FOR ARF2
127	CHECK	BTST	D2, D0	CHECK IF REPORTING ENABLED
128		BEQ.S	DISARF	DISABLE THE REPORTING
129		MOVE.W	#ENABLE, DFRPEN	ENABLE REPORTING
130		BRA.S	CHKAC	
131	DISARF	CLR.W	DFRPEN	DISABLE REPORTING
132	CHKAC	MOVE.B	DF_ASUB(A2), D0	GET SUBTYPE
133		BTST	#DF_AUDIO, D0	CHECK IF ACORR TUNE
134		BEQ	NONE	DO NOT TUNE, FINISHED
135		LEA	DF_FR1(A2), A0	GET POINTER TO 1ST BYTE OF FREQ
136		LEA	TSTRING, A1	GET POINTER TO OUTPUT BUFFER
137		JSR	FMTRFQ	FORMAT FREQUENCY

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFAL.S68 NUM

```

138 MOVE.B DF_RNG(A2),D0      GET BAND WIDTH AND DETECTOR
139 BSET #AGCTC,D0           MAKE SURE SLOW TIME CONST.
140 BCLR #ACCDUMP,D0         REQUEST AGC DUMP
141 MOVE.W D0,RNG            PUT INTO TUNING COMMAND
142 MOVE.B DF_FRI(A2),D1     GET MSB OF FREQ.
143 CLR.W GHZ               CLEAR GHZ FLAG
144 AND.W #GHZFELD,D1       MASK OFF GHZ FIELD
145 BEQ.S NOGHZ             NO GHZ FIELD
146 MOVE.W #1,GHZ           ELSE, SET GHZ FLAG
147 ?WRITW DSOB,#N,RUFAD...D0
148 BSET #ACCDUMP,D0        REQUEST AGC ATTACK
149 MOVE.W D0,RNG           PUT BACK INTO TUNING COMMAND
150 *
151 * SPIN HERE FOR A WHILE
152 * WHILE AGC DUMP:
153 *
154 MOVE.W #COUNT,D1       GET LOOP COUNT
155 DBF D1,WAIT            SPIN LOOP
156 *
157 ?WRITW DSOB,#N,RUFAD
158 *
159 DONE DFA2,#2,#0,#1,IPACK ; SCHEDULE DFA2
160 *
161 ?EXIT
162 *
163 END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000_DF] DFA2.S68 NUM

19.2 DFA2.S68_NUM

***** Source Listing ==> DFA2.S68 NUM *****

```

1  *      TTL      'DFA2'
2  *
3  *
4  *
5  *      VERSION  DATE
6  *      1.0      7/11/83
7  *      1.1      1/16/84  PYW - INSERT TWO TIMING BITS
8  *
9  *SINCLUDE DFA2.FMT/G
10 *
11 *      I/O DEVICES USED:
12 *      FAST DF CONTROLLER
13 *      SIO - 3 CHANNEL A      RF PROCESSOR
14 *      SIO - 3 CHANNEL B      DIAL CHANNEL (DF) RECEIVER
15 *      PIO                    ANTENNA CONTROL UNITS
16 *
17 *
18 *      ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
19 *      ADFUC
20 *
21 *      METHOD:
22 *
23 * PDL:
24 *
25 * ?DLINK IPCK      /* GET DF REQUEST PACKET */
26 * DFER <- NONE     /* INITIALIZE ERROR FLAG */
27 * TUNED <- FALSE    /* INITIALIZE TUNED RCVR FLAG */
28 * CALIB <- FALSE    /* INITIALIZE CALIB 1ST SEGMENT FLAG */
29 * ADFBT <- NOT ADF BITE /* INITIALIZE ADF BITE FLAG */
30 * SIGNAL <- NOT PRESENT /* INITIALIZE SIGNAL PRESENT FLAG */
31 * IPACK <- NULL     /* INDICATE NO RESULTS PACKET */
32 * GET DF TYPE      /* FROM IPCK */
33 * IF TYPE = BITE
34 * THEN /* CHECK FOR ADF BITE */
35 * IF SUBTYPE = COARSE /* CHECK FOR ADF BITE */
36 * THEN /* DETERMINE IF ON, OFF, OR NORMAL */
37 * CALL GETBITS(SUBTYPE, DF_SEG) /* GET SEGMENT NO. */
38 * IF SEG NO. = ADF BITE ON
39 * THEN /* INDICATE ADF BITE ON */
40 * PTR <- ADF BITE ON SEQ. HEADER /* GET SEQUENCE HEADER */
41 * ADFBT <- ON/OFF
42 * FALSE /* CHECK IF ADF BITE OFF */
43 * IF SEG NO. = ADF BITE OFF

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFA2.S68 NUM

```

44 *      THEN /* INDICATE ADF BITE OFF */
45 *      ITR <- ADF BITE OFF SEQ. HEADER
46 *      ADFBT <- ON/OFF
47 *      ELSE /* NOT ON OR OFF, NORMAL DATA COL. BITE */
48 *      ADFBT <- NORMAL
49 *      GO TO FINDPTR /* NEED TO GET SEQ. HEADER */
50 *      ENDIF
51 *
52 *      ELSE /* TYPE <> BITE */
53 *      FINDPTR: CALL DETBAND(DIFREQ, BAND) /* DETERMINE FREQ BAND */
54 *      CALL GETPACK(IPACK) /* GET A PACKET FOR RESULTS */
55 *      FORMAT PACKET WITH DF IDENTIFIERS
56 *      GET POINTER TO BASE OF ACCUMULATORS IN PACKET
57 *      CALL CLRACCS(ACCPTR) /* CLEAR OUT ACCUMULATORS */
58 *      IF COARSE DF REQUEST
59 *      THEN /* CHECK IF ADF BITE */
60 *      IF ADFBT = NORMAL
61 *      THEN /* GET PSUEDO SEG. NO. FOR ADF BITE */
62 *      SEGNO <- ADF BITE SEQ. NO.
63 *      ELSE /* NORMAL COARSE DF REQUEST */
64 *      SEGNO <- COARSE DF SEQ. NO. /* PSUEDO FOR COARSE */
65 *      ENDIF
66 *      ELSE /* FINE REQUEST */
67 *      CALL GETBITS(DF_SEG, SEGNO) /* GET ACTUAL SEGMENT NO. */
68 *      SIGNAL <- TRUE /* ASSUME SIGNAL PRESENT FOR FINE */
69 *      IF TYPE = CALIB
70 *      THEN /* CHECK IF CALIB SEGMENT NO. 1 */
71 *      IF SEGNO = CALIB SEGMENT NO. 1
72 *      THEN /* SET FLAG */
73 *      CALIB <- TRUE
74 *      ENDIF
75 *      ENDIF
76 *
77 *      ENDIF
78 *      BASED ON BAND AND SEGNO, GET PTR TO SEQUENCE HEADER
79 *      ENDIF
80 *      CALL INITDF /* INITIALIZE FDPC */
81 *      GET NUM /* FROM DATA COL. SEQ. HEADER */
82 *      GET RFPFTR /* FROM DATA COL. SEQ. HEADER */
83 *      GET ACUPTR /* FROM DATA COL. SEQ. HEADER */
84 *      GET FDPCPTR /* FROM DATA COL. SEQ. HEADER */
85 *      IF NEED TO TUNE RECEIVER /* CHECK SUBTYPE */
86 *      THEN /* GET POINTERS TO TUNING COMMAND SEQUENCES */
87 *      TRFPFTR <- TUNING SEQ. RF PROC. COMMANDS /* BASED ON BAND */
88 *      TACUPTR <- TUNING SEQ. ACU COMMANDS /* BASED ON BAND */
89 *      TDFPTR <- TUNING SEQ. FDPC COMMAND /* BASED ON BAND */
90 *      CALL FMTRFQ(DF_FREQ, OUTREQ) /* REFORMAT FREQUENCY */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFA2.S68.NUM

```

91 * CALL SILOAD(OUTPRE) /* DUMP AGC */
92 * CALL SISEND(CHANL, B) /* ENABLE CHANNEL B OUTPUT */
93 * CALL RFPSETUP(RFPFTR) /* LOAD RF PROC. COMMANDS */
94 * CALL SISEND(CHANL, A) /* ENABLE CHANNEL A OUTPUT */
95 * CALL ACUSEFUP(ACUPTR) /* LOAD ACU COMMANDS */
96 * CALL STARTUP(TDFPCTR) /* INITIATE TUNING STEP */
97 * CALL GETBITS(FDFCCOM, FD MODE, MODE) /* GET MODE */
98 * COUNT <- 1 /* INITIALIZE STEP COUNT */
99 * CALL NQUEDE(RFPFTR, COUNT, MODE) /* QUEUE ITEMS */
100 * TUNED <- TRUE /* TUNED RCVR FLAG */
101 * IF FINE DF /* CHECK SUBTYPE */
102 * THEN /* SKIP OVER 1ST STEP */
103 * RFPFTR <- RFPFTR + <STEP SIZE> /* JUMP TO NEXT STEP */
104 * ACUPTR <- ACUPTR + <STEP SIZE> /* JUMP TO NEXT STEP */
105 * FDFCCTR <- FDFCCTR + <STEP SIZE> /* JUMP TO NEXT STEP */
106 * ENDF
107 * ELSE /* DO NOT NEED TO TUNE THE RECEIVER */
108 * CALL RFPSETUP(RFPFTR) /* LOAD RF PROC. COMMANDS FOR 1ST STEP */
109 * CALL SISEND(CHANL, A) /* ENABLE CHANNEL A OUTPUT */
110 * CALL ACUSEFUP(ACUPTR) /* LOAD ACU COMMANDS FOR 1ST STEP */
111 * CALL STARTUP(FDFCCTR) /* INITIATE 1ST COLLECTION STEP */
112 * CALL GETBITS(FDFCCOM, FD MODE, MODE) /* GET MODE OUT OF FDFC COM. */
113 * COUNT <- 1 /* INITIALIZE COUNT */
114 * CALL NQUEDE(RFPFTR, COUNT, MODE) /* QUEUE ITEMS */
115 * ENDF
116 * CALL GETBITS(DF BAND, BNDWDTH) /* GET BANDWIDTH CODE */
117 * CALL PUTPAK(IPCK) /* RETURN INPUT PACKET */
118 *
119 * IF COUNT=NUM THEN GO TO DONE /* ONLY ONE STEP */
120 *
121 * LOOP I = 2 TO N'M /* LOOP THROUGH REMAINING STEPS */
122 * CALL RFPSETUP(RFPFTR) /* LOAD A STEP OF RF PROC. COMMANDS */
123 * CALL ACUSEFUP(ACUPTR) /* LOAD A STEP OF ACU COMMANDS */
124 * ENDOOP
125 * CALL SISEND(RFPFTR) /* ENABLE RF PROC. COMMAND OUTPUT */
126 * ABRT <- FALSE /* RESET NO SIGNAL ABORT FLAG */
127 *
128 * -----
129 *
130 * DF COLLECTION LOOP
131 *
132 * -----
133 * DELOOP : COUNT <- COUNT + 1 /* INCR. STEP COUNT */
134 * IF COUNT > NUM THEN GO TO DONE /* EXIT LOOP TEST */
135 * GET NEXT FDFC COMMAND /* FROM FDFCCTR */
136 * CALL ADJFDFC(INF, FC) /* ADJUST FDFC COMMAND */
137 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA2.S68 NUM

```

138 * CALL STARTDF(NF,FC) /* BUFFER THE FDPC COMMAND */
139 *
140 * CALL GETBITS(FD MODE,MODE) /* GET DATA COLLECTION MODE */
141 * CALL NOQUEUE(RFPFTR,COUNT,MODE) /* ENQUEUE THE ITEMS */
142 * RFPFTR <- RFPFTR + <SIZE OF COMMAND STEP>
143 *
144 *
145 * CALL WAITDF /* WAIT FOR FIRST DF */
146 *
147 * CALL DOQUEUE(RFPFTR,PCOUNT,PMODE) /* GET ASSOCIATED PARAMS. */
148 * IF DID NOT TUNE RCVR /* CHECK TUNED FLAG */
149 * THEN
150 * IF NOT CALIB SEGMENT NO. 1 /* CHECK CALIB FLAG */
151 * THEN
152 * IF ABRT = FALSE /* SIGNAL IS PRESENT */
153 * THEN
154 * IF SIGNAL NOT PRESENT /* CHECK SIGNAL FLAG */
155 * THEN /* CHECK FOR SIGNAL PRESENCE */
156 * CALL DMFIFO(BUFFER) /* EMPTY DATA FIFO INTO WORK AREA
157 * CALL SIGPRES(BUFFER,RFPFTR,OPPTR,OPPTR,N) /* CHECK FOR
158 * SIGNAL PRESENCE */
159 * IF SIGNAL PRESENT /* CHECK SIGNAL FLAG AGAIN */
160 * THEN /* GET THE GOOD DATA */
161 * LOOP I = 1 TO N
162 * CALL ACCBUF(ACCUMS,OPPTR,OPPTR) /* ACCUMULATE
163 * ENDLOOP
164 * ELSE /* NO NEED TO COLLECT MORE DATA */
165 * ABRT <- TRUE /* SET NO SIGNAL FLAG */
166 *
167 * ENDIF
168 * ELSE /* SIGNAL IS PRESENT */
169 * GET N /* FROM RFPFTR */
170 * LOOP I = 1 TO N
171 * CALL ACCUM(ACCUMS,RFPFTR,PMODE) /* ACCUMULATE DATA
172 * ENDLOOP
173 * ENDIF /* SIGNAL IS PRESENT */
174 * ELSE /* ABRT = TRUE */
175 * CALL DMFIFO(BUFFER) /* THROW OUT DATA */
176 * /* ABRT = FALSE */
177 * ELSE /* CALIB SEGMENT NO. 1 */
178 * IF PCOUNT = STEP X
179 * THEN /* TAKE AGC MEASUREMENT, NOISE */
180 * CALL READAGC(AGCVAL) /* READ AGC */
181 * SD_SSM <- AGCVAL /* STORE INTO CURRENT DF */
182 * ELSE /* CHECK IF NEED TO TAKE 2ND MEASUREMENT */
183 * IF PCOUNT = STEP Y
184 * THEN /* TAKE AGC MEASUREMENT, SIGNAL */
185 * CALL READAGC(AGCVAL) /* READ AGC */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.S68_NUM

```

185 *          SD_SSM <- ACQVAL - SD_SSM /* COMPUTE DIFFERENCE */
186 *          ENDIF
187 *          ENDIF /* PCOUNT = STEP X */
188 *          ENDIF /* CALIB SE/MENT NO. 1 */
189 *          ELSE /* DID NOT TUNE RCVR */
190 *          CALL DMFF:FO(BUFFER) /* THROW DATA AWAY */
191 *          TUNED <- FALSE /* RESET FLAG */
192 *          ENDIF
193 *
194 *          BOTTOM : GO TO DFLOOP /* END OF DF LOOP */
195 *
196 *          -----
197 *
198 *          END OF DF COLLECTION LOOP
199 *
200 *          -----
201 *
202 *          DONE : CALL WAITDF /* EMPTY PIPE */
203 *
204 *          CALL DQUEUE(PRFPTR,PCOUNT,RMODE) /* DEQUEUE LAST SET OF ID */
205 *          IF NOT ADF BITE ON OR OFF
206 *          THEN /* CHECK IF SIGNAL PRESENT */
207 *          IF SIGNAL IS PRESENT /* CHECK SIGNAL FLAG */
208 *          THEN /* ACCUMULATE DATA */
209 *          CALL ACCUM(ACCUMS,PRFPTR,RMODE) /* ACCUMULATE DATA */
210 *          ELSE /* FLAG ERROR */
211 *          DFERR <- NO SIGNAL
212 *          ENDIF
213 *          SD_ERR <- DFERR /* PUT DF ERRORS IN CURRENT DF */
214 *          IF_FINE DF SEGMENT /* CHECK SUBTYPE */
215 *          THEN /* SCHEDULE THE ACCUMULATE FOR FINE JOB */
216 *          ?SCHED DFA5,#2,...,IPACK
217 *          ELSE /* SCHEDULE THE LOB CALC. JOB */
218 *          ?SCHED DFA6,#2,...,IPACK
219 *          ENDIF
220 *          ENDIF
221 *          DONE WITH CURRENT <- TRUE /* SET FLAG TO INDICATE DONE */
222 *
223 *          ?EXIT
224 *
225 *          -----
226 *
227 *
228 *          DFA2 IDNT
229 *          OPT M,-X
230 *          XDEF DFAZE
231 *          XDEF NPEND

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFA2.S68 NUM

232	XDEF	NCOMPL
233	XDEF	RESET
234	XDEF	RESOVF
235	XDEF	WAITING
236	XDEF	DFBOX
237	XDEF	RMSG
238	XDEF	RID
239	XDEF	RDATA,TDATA
240	XDEF	SENDID
241	XDEF	BAND
242	XREF	DFDIB
243	XDEF	REFLDPOS
244	XDEF	RFPFTR
245	XDEF	BNDWOTH
246	XDEF	ADFBT
247	XDEF	SIGNAL
248	XDEF	SKFRST
249	*	DMCUR,DFERR,NAVDATA
250	*	
251	XDEF	INTCPU
252	*	
253	*	
254	*	SUBROUTINES
255	*	
256	XREF	GETBITS
257	XREF	DET BAND
258	XREF	GETLPAK,PUTPAK
259	XREF	CLRACCS
260	XREF	INITDF,STARTDF,WAITDF
261	XREF	FMTRQ
262	XREF	SILOAD,SIOSEND
263	XREF	REPSETUP,ACUSETUP
264	XREF	NQUEUE,DQJUE
265	XREF	ADJDFC
266	XREF	DMFFIFO,ACCBUF,ACCUM
267	XREF	SIGPRES
268	XREF	DETSIG
269	XREF	READAGC
270	*	
271	*	EXTERNAL VARIABLES (OR TABLES)
272	*	
273	XREF	ABTON,ABTOFF
274	XREF	SEQTAB
275	XREF	SEQTAB2
276	XREF	TUNESQ,TUNEACU,TUNEFDFC
277	XREF	DFA5,DFA6
278	XREF	DFA2DAT1 ; ADDR OF TIMING WORD #1

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO: IGR.MC68000_DF]DFA2.S68_NUM

279 XREF DFA2OUT1 ;ADDR TO OUTPUT TIMING WORD #1 TO
 280 XREF DFA2MSK1 ;MASK USED TO SET/CLEAR TIMING BIT #1
 281 XREF DFA2DAT2 ;ADDR OF TIMING WORD #2
 282 XREF DFA2OUT2 ;ADDR TO OUTPUT TIMING WORD #2 TO
 283 XREF DFA2MSK2 ;MASK USED TO SET/CLEAR TIMING BIT #2
 284 *

285 XREF NAVDATA CURRENT NAVDATA
 286 XREF CALIB CALIB SEC. 1 FLAG
 287 *

288 *SINCLUDE ARTEMACS.S/G NOLIST, ?OLINK, ?SCHED, ?EXIT
 289 *

290 *SINCLUDE ARTEDATA.S/G NOLIST,RCB
 291 *

292 *SINCLUDE STRC.MAC/S
 293 *

294 *SINCLUDE DFC.DBS/G
 295 *

296 *SINCLUDE DF.PAC/G
 297 *

298 *SINCLUDE SD.PAC/G
 299 *

300 *SINCLUDE DFTYPE.EQU/G
 301 *

302 *SINCLUDE DFERRS.EQU/G
 303 NOLIST

304 *SINCLUDE MIDEQU.EQU/G
 305 LIST
 306 *

307 * FD FAST DF CONTROLLER COMMAND REGISTER
 308 *

309 STRC
 310 USES 2,FD,COM
 311 BITS 4,FD,ACU
 312 BITS 4,FD,RFP
 313 BITS 2,FD,RCVR
 314 BITS 2,FD,MODE
 315 BITS 2,FD,DLY
 316 BITS 1,FD,TEST
 317 BITS 1,FD,NORM
 318 ENDST FD
 319 *

320 * PARALLEL OUTPUT REGISTERS
 321 *

322 * ORG .FFFFFF488
 323 DATA DS.W 1
 324 DATA DS.W 1
 325 DATA DS.W 1

DATA CHANNEL A
 DATA CHANNEL B

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFA2.S68_NUM

```

326 PCNTL DS W 1 CONTROL REGISTER
327 PSTAT DS W 1 STATUS REGISTER
328 *
329 *
330 *
331 SECT ADFUC
332 *
333 * variables indicating state of DF driver
334 *
335 MPEND: DS.W 1 *number of outstanding DF's (max=2)
336 MCOMPL: DS.W 1 *number of completed DF's (max=)
337 RESET: DS.B 1 *logical set by resetdf
338 RESOVF: DS.B 1 *logical indicating result queue overflow
339 WAITING DS.B 1 *indicates that task awaits result
340 *
341 * IDENTIFIERS OF CURRENT DF
342 *
343 TUNED DS.W 1 TUNED THE RCVR
344 ADFBT DS.W 1 ADF BITE ON-OFF, NORMAL, OR NOT
345 SIGNAL DS.W 1 SIGNAL PRESENT OR NOT
346 PIPE DC W 1 PIPELINE MODE
347 ABRT DS.W 1 ABORT IF NO SIGNAL FLAG
348 SKFRST DS.W 1 SKIP A BL MSRMNT FLAG
349 *
350 * declarations concerning inbound and outbound mail
351 *
352 DFBOX: DS.W 1 *boxid for dfcomplete message values= null,cle*
353 RMSG: DS.L 0 *place to receive message
354 RID: DS.W 1 *transmitter id
355 RDATA: DS.L 1 *received data
356 TDATA: DS.L 1 *transmitted data
357 SENDID: DS.L 1 *sender id
358 *
359 BAND DS.W 1 FREQUENCY BAND CODE
360 RFPLDPOS DS.W 1 SIO PTR FOR SIO - 3 (A)
361 RFPPTR DS.L 1 RF PROC COMMAND PTR
362 *
363 EMPTY: EQU 0 *indicates null message queue
364 *
365 *df returned message types
366 *
367 RESULT EQU 0 *normally returned DF message
368 CLEAR EQU 1 *indicates need to dump message queue
369 *
370 *error returns from startdf and waitdf
371 *
372 NONE: EQU 9 *no error

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFA2.S68_NUM

373	STOVF:	EQU	1	*start overflow
374	ABORT:	EQU	2	*aborted by call to resetdf
375	ROVF:	EQU	3	*too many completed results in fifo
376	MAXCOMP	EQU	2	*maximum number of lts allowed to accumula*
377	*			
378	ADON	EQU	2	ARF DF BITE ON FLAG
379	ADOFF	EQU	3	ARF DF BITE OFF FLAG
380	ONOFF	EQU	-1	ADF BITE ON OR OFF FLAG
381	YES	EQU	1	
382	CAL1	EQU	6	CALIB SEGMENT NO. 1 CODE
383	BITESEG	EQU	8	ADF BITE SEGMENT NO.
384	COARSEG	EQU	7	COARSE DF SEGMENT NO.
385	BANDSEQ	EQU	8*DFC	SIZE OF ONE TABLE
386	TRUE	EQU	1	
387	*			
388	TWOMEAS	EQU	16	2 RAW DATA MSRMNT SETS (A & C)
389	ONEMEAS	EQU	8	1 RAW DATA MSRMNT SET (A & C)
390	*			
391	SIORINIT	EQU	\$0	CHANNEL B START ADDR.
392	SIOAINIT	EQU	\$800	CHANNEL A START ADDR.
393	*			
394	STEPX	EQU	2	CALIB SEGMENT STEP 1
395	STEPE	EQU	4	CALIB SEGMENT STEP 2
396	NOILEV	DC.W	\$53	AGC VALUE AT -112 DEM
397	*			
398	RFPTUNE	EQU	6*2	SIZE OF RFP TUNING SEQ.
399	ACUTUNE	EQU	2*2	SIZE OF AGU TUNING SEQ.
400	*			
401	IPACK	DS.L	1	PACKET POINTER (SD)
402	BNWDTH	DS.W	1	BANDWIDTH
403	*			
404	FREQBUF	DS.W	1	1ST 4 DIGITS OF FREQ.
405		DS.W	1	2ND 4 DIGITS OF FREQ.
406		DC.W	\$00C0	AGC SELECT, LMGC, MCC WEIGHT
407	BW	DS.W	1	BANDWIDTH, DETECTOR, AGC CONTROL
408	GHZ	DS.W	1	GHZ FLAG
409		DC.W	0	
410	*			
411	FREQFMT	DC.W	\$0010	
412		DC.W	\$0010	
413		DC.W	\$0008	
414		DC.W	\$0008	
415		DC.W	\$0006	
416		DC.W	\$0002	
417	*			
418	NFREQ	EQU	6	NO. OF DATA/FORMAT ELEMENTS
419	*			

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.S68_NUM

420	FDECOM	DS L	1	HOLDER FOR FDFC COMMAND PTR
421	ACCBASE	DS L	1	BASE OF ACCUMULATOR
422	*			
423	BUFFER	DS W	64	TEMPORARY BUFFER
424	*			
425	*			
426	ACCOUNT	DC L	0	COUNT OF ENTRIES
427	INTCPU	DC W	0	
428	*			
429	*			
430	*			
431	* DFA2 - DF DATA COLLECTION			
432	*			
433	*			
434	*			
435	DFA2E	?DLINK A1		GET PACKET POINTER
436	*			
437		MOVEA.L (DFA2DAT1),A3		ADDR OF TIMING WORD #1
438		MOVEA.L (DFA2OUT1),A4		ADDR TO OUTPUT TIMING WORD #1 TO
439		MOVE.W (DFA2MSK1),D3		MASK USED TO SET TIMING BIT #1
440		OR.W D3,(A3)		SET TIMING BIT #1
441		MOVE.W (A3),(A4)		OUTPUT TIMING WORD #1
442		MOVEA.L (DFA2DAT2),A3		ADDR OF TIMING WORD #2
443		MOVEA.L (DFA2OUT2),A4		ADDR TO OUTPUT TIMING WORD #2 TO
444		MOVE.W (DFA2MSK2),D3		MASK USED TO SET TIMING BIT #2
445		OR.W D3,(A3)		SET TIMING BIT #2
446		MOVE.W (A3),(A4)		OUTPUT TIMING WORD #2
447		MOVE.W #1,INTCPU		
448		ADDQ.L #1,ACCOUNT		INCREMENT ENTRY COUNTER
449	*			
450	*			
451	* INITIALIZE FLAGS			
452	*			
453	*	CLR.W DFERR		INITIALLY NO ERRORS
454		CLR.W TUNED		RESET TUNED RCVR FLAG
455		CLR.W CALIB		RESET CALIB SEG 1 FLAG
456		CLR.W ADPBT		RESET ADF BITE FLAG
457		CLR.W SIGNAL		RESET SIGNAL PRESENT FLAG
458		CLR.L IPACK		INDICATE NO PACKET
459		CLR.W SKFRST		RESET SKIP MSGMNT FLAG
460	*	CLR.W DFERR		RESET TOOK TOO LONG FLAG
461	*			
462	* CHECK FOR ADF BITE			
463	*			
464		MOVE.B DF_TYP(A1),D0		GET DF TYPE
465		CMF.B #TBITE,D0		CHECK IF BITE
466		BNE.S FINDPTR		NOT BITE, CONTINUE

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFA2.S68_NUM

```

467 MOVE.B DF_ASUB(A1),D0 GET SUBTYPE
468 BTST #DF_FINE,D0 CHECK IF ADF BITE
469 BNE.S FINDPTR BITE, BUT FINE
470 *
471 * ADF BITE, CHECK IF ON, OFF, OR NORMAL
472 *
473 MOVE.W #DF_SEG,D1 GET DESCRIPTOR
474 JSR GETBITS EXTRACT SEGMENT NUMBER
475 CMP.W #ADFON,D1 CHECK IF ADF BITE ON
476 BNE.S CHKOFF NOT ON, SEE IF OFF
477 LEA ABTON,A2 GET PTR TO ADF BITE ON SEQ
478 MOVE.W #ONOFF,ADFBT SET FLAG TO ON/OFF
479 BRA STRICOL READY TO START
480 *
481 CHKOFF CMP.W #ADFOFF,D1 CHECK IF ADF BITE OFF
482 BNE.S NORMBIT NOT ON OR OFF, NORMAL ADF BITE
483 LEA ABTOFF,A2 GET PTR TO ADF BITE OFF SEQ
484 MOVE.W #ONOFF,ADFBT SET FLAG TO ON/OFF
485 BRA STRICOL READY TO START
486 *
487 NORMBIT MOVE.W #YES,ADFBT NORMAL ADF BITE
488 *
489 *****
490 *
491 * GET POINTER TO APPROPRIATE DATA COLLECTION SEQUENCE
492 *
493 *****
494 *
495 FINDPTR LEA DF_FR1(A1),A0 GET PTR OF FREQ STRING
496 JSR DETBAND DETERMINE FREQUENCY BAND
497 MOVE.W D0,BAND SAVE BAND CODE
498 MOVE.W D0,D2 SAVE BAND
499 MOVE.B DF_RNG(A1),D0 GET OVERALL FIELD
500 MOVE.W #DF_BAND,D1 GET DESCRIPTOR
501 JSR GETBITS GET BAND WIDTH CODE
502 MOVE.W D1,BNDWDTH SAVE BANDWIDTH CODE
503 *
504 * PREPARE A LARGE PACKET FOR RESULTS
505 *
506 JSR GETLPK GET A LARGE PACKET
507 MOVE.B D2,SD_BAND(A6) SET FREQ BAND CODE
508 MOVE.B DF_TYP(A1),SD_TYP(A6) DF TYPE
509 MOVE.W DF_ACC(A1),SD_ACC(A6) ACCOUNTABILITY
510 MOVE.B DF_ASUB(A1),SD_ASUB(A6) SUBTYPE
511 MOVE.B DF_FR1(A1),SD_FR1(A6) 1ST BYTE OF FREQ
512 MOVE.B DF_FR2(A1),SD_FR2(A6) 2ND BYTE OF FREQ
513 MOVE.B DF_FR3(A1),SD_FR3(A6) 3RD BYTE OF FREQ

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF|DFA2.S68_NUM

```

514 MOVE.B DF_FR4(A1),SD_FR4(A6) 4TH BYTE OF FREQ
515 MOVE.B DF_FR5(A1),SD_FR5(A6) 5TH BYTE OF FREQ
516 CLR.W SD_ERR(A6) INITIALIZE ERROR CODE (NONE)
517 MOVE.L A6,IPACK SAVE PACKET ADDRESS
518 LEA SD_BALDI(A6),A0 GET ADDR. OF ACCUMULATORS
519 MOVE.L A0,ACCBASE SET ACCUMULATOR ADDR.
520 JSR CLRACCS CLEAR ACCUMULATORS
521 *
522 * GET SEGMENT NUMBER (OR PSUEDO SEGMENT NUMBER)
523 *
524 MOVE.B DF_ASUB(A1),D1 GET SUBTYPE
525 BTST #DF_FINE,D1 CHECK IF COARSE OR FINE
526 BNE.S FINE IF 1 THEN FINE
527 TST.W ADFBT CHECK IF ADF BITE (NORMAL)
528 BEQ.S COARSE NOT ADF BITE, CONTINUE
529 MOVE.W #BITESeg,D2 SET INDEX TO BITE SEGMENT
530 BRA.S GETSEQ GO GET SEQUENCE PTR
531 *
532 COARSE MOVE.W #COARSESeg,D2 SET INDEX TO COARSE SEGMENT
533 BRA.S GETSEQ GO GET SEQUENCE PTR
534 *
535 FINE MOVE.B DF_ASUB(A1),D0 GET OVERALL FIELD
536 MOVE.W #DF_SEG,D1 GET DESCRIPTOR
537 JSR GETBITS GET SEGMENT NUMBER
538 MOVE.W D1,D2 SET SEGMENT NUMBER
539 MOVE.W #TRUE,SIGNAL ASSUME SIGNAL PRESENT FOR FINE
540 CMP.B #TCALIB,DF_TYP(A1) CHECK IF CALIB
541 BNE.S GETSEQ IF NOT, CONTINUE
542 CMP.W #CAL1,D1 CHECK IF CALIB SEG. NO. 1
543 BNE.S GETSEQ IF NOT, CONTINUE
544 MOVE.W #TRUE,CALIB ELSE SET FLAG
545 *
546 * BASED ON BAND AND SEGMENT NUMBER, GET
547 * COLLECTION SEQUENCE HEADER
548 *
549 GETSEQ MOVE.W BAND,D0 GET BAND CODE
550 SUBQ.W #1,D0 ADJUST FOR INDEX
551 MULS #BANDSEQ,D0 MULTIPLY BY TABLE ENTRY
552 LEA SEQTAB,A2 GET ADDRESS OF SEQUENCE TABLE
553 CMP.B #TCALIB,DF_TYP(A1) CHECK IF DOING A CALIB
554 BNE.S USETAB1 NOT CALIB, USE ORIGINAL TABLE
555 MOVE.B DF_FLAGS(A1),D1 GET MISC. FLAG FIELD
556 BTST #DF_CV,D1 CHECK IF COARSE VERIFY
557 BEQ.S USETAB1 NOT COARSE VERIFY, USE OR. TABLE
558 LEA SEQTAB2,A2 COARSE VERIFY, USE OTHER TABLE
559 *
560 USETAB1 ADD.W D0,A2 ADJUST BY OFFSET

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFA2.S68_NUM

```

561      SUBQ W #1,D2      ADJUST INDEX FOR OFFSET
562      MULLS #DFC,D2      MULTIPLY BY SUB-ENTRY
563      ADD W #2,A2      GET PTR TO SEQUENCE
564      *****
565      *****
566      *****
567      *****
568      *****
569      *****
570      *****
571      *****
572      STRTOL
573      JSR INITDF      RESET THE FDFC
574      MOVE W DFC_NUM(A2),D6      GET NUMBER OF STEPS
575      BEQ FINIO      IF NOTHING TO DO, EXIT
576      MOVE L DFC_RFP(A2),A6      GET RFP COMMAND PTR
577      MOVE L DFC_ACU(A2),A5      GET ACU COMMAND PTR
578      MOVE L DFC_FDFC(A2),A4      GET FDFC COMMAND PTR
579      *****
580      *****
581      *****
582      *****
583      *****
584      *****
585      *****
586      *****
587      *****
588      *****
589      *****
590      *****
591      *****
592      *****
593      *****
594      *****
595      *****
596      *****
597      *****
598      *****
599      *****
600      *****
601      *****
602      *****
603      *****
604      *****
605      *****
606      *****
607      *****

```

HAVE PTR TO COLLECTION SEQUENCE (A2).
 START COLLECTION
 DETERMINE IF NEED TO TUNE RCVR OR NOT
 MOVE B DF_ASUB(A1),D0 GET SUBTYPE
 BNE NOTUNE CHECK IF NEED TO TUNE
 IF CONTIGUOUS, NO TUNE
 SET UP TUNING STRINGS FOR DF RECEIVER
 MOVE L A1,A3 SAVE A COPY OF IPACK
 LEA FREQBUF,A1 GET PTR TO OUTPUT BUFFER
 LEA IF_FR1(A3),A0 GET PTR TO BCD STRING
 JSR FMTRQ FORMAT 5 BCD TO 4 BCD
 CLR W #0 CLEAR GHZ FLAG
 MOVE B DF_FR1(A3),D0 GET 1ST BYTE OF FREQ
 AND W #3F0,D0 MASK OFF GHZ NIBBLE
 BEQ S_NOGHZ IF 0 THEN NO GHZ
 MOVE W #1,GHZ ELSE SET GHZ BIT
 MOVE B DF_RNG(A3),D0 GET BANDWIDTH AND DETECTOR
 BSET #DF_TC,D0 REQ SLOW AGC TIME CONST.
 BCLR #DF_DU,D0 REQ AGC DIMP
 NOTE
 THE RECEIVER WILL GENERATE A TONE AT EVERY FREQUENCY
 IF THE DETECTOR TYPE SELECTED IS CW THE TONE IS FOR
 AUDIO MONITORING OF SIGNALS. HOWEVER THE TONE WILL
 CAUSE IN-CORRECT DETECTIONS OF SIGNALS (I.E. MISINTERPRETING
 THE TONE AS A POSSIBLE SIGNAL). FORCE THE DETECTOR
 TYPE TO IM (CODE 000) TO PREVENT GENERATION OF THE TONE.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA2.S68_NUM

```

608 ANDI.L #FFFFFF,D0      force detector type to FM
609 MOVE.W #0,BW          SET IN DATA BUFFER
610 LEA   FREQMT,A0       GET PTR TO FORMAT LIST
611 MOVE.W #FREQ,D0       GET NUMBER OF ELEMENTS
612 SWAP  D0              GET NUMBER IN UPPER WORD
613 CLR.W D1              CLEAR SIO PTR
614 JSR   SIOLOAD         LOAD 1ST COMMAND (DUMP AGC)
615
616 MOVE.W #SIOINIT,D0     GET START ADDR. FOR RCVR CHANNEL
617 JSR   SIOSEND         PREPARE SIO FOR OUTPUT
618
619 * SET UP RF PROC COMMANDS FOR TUNING
620
621 LEA   TUNSEQ,A0        GET PTR TO RFP TUNING SEQ
622 MOVE.W BAND,D1         GET BAND CODE
623 SUBQ.W #1,D1          DECR. FOR OFFSET
624 MOVE.W D1,D2           GET A COPY
625 Muls  #RFPTUNE,D1      MULTIPLY BY SIZE OF ENTRY
626 ADD.W D1,A0            ADJUST ADDR. BY OFFSET
627 MOVE.L A0,REPTR        SET REPTR TO TUNING SEQ
628 MOVE.L A0,A2           GET A COPY OF REPTR
629 CLR.W RFPDPOS          RESET SIO ADDR. PTR
630 JSR   RFPSETUP         LOAD SIO WITH RFP COMMANDS
631 MOVE.W #SIOAINIT,D0    GET START ADDR. FOR RFP CHANNEL
632 JSR   SIOSEND         PREPARE SIO FOR OUTPUT
633
634 * SET ACU COMMANDS FOR TUNING
635
636 MOVE.W D2,D1           GET INDEX
637 Muls  #ACUTUNE,D1      MULTIPLY BY SIZE OF ENTRY
638 LEA   TUNEACU,A0       GET ADDR. OF ACU TUNIG TABLE
639 ADD.L D1,A0            ADJUST ADDR. BY OFFSET
640 JSR   ACUSETUP         LOAD PIO WITH ACU COMMANDS
641
642 * GET FDFC COMMAND FOR TUNING SEQUENCE
643
644 ASL.W #2,D2            MULTIPLY INDEX BY OFFSET (4)
645 LEA   TUNEFDFC,A0      GET TABLE POINTER
646 MOVE.L 0(A0,D2.W),D0   GET FDFC COMMAND
647 JSR   ADJFDFC          ADJUST FDFC COMMAND
648
649 * INITIATE TUNING SEQUENCE
650
651 JSR   STARTDF          START DF SEQUENCE
652
653 MOVEM.L D0/A0-A1,-(SP)  SAVE REGISTERS USED FOR TIMING
654 MOVEA.L (DFA2DAT2),A0  ADDR OF TIMING WORD #2

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFA2.S68_NUM

```

655 MOVE.L (DFA2OUT2),A1      ADDR TO OUTPUT TIMING WORD #2 TO
656 MOVE.W (DFA2MSK2),D0      MASK USED TO CLEAR TIMING BIT #2
657 FOR W D0,(A0)             CLEAR TIMING BIT #2
658 MOVE.W (A0),(A1)          OUTPUT TIMING WORD #2
659 MOVEM.L (SP)+,D0/A0-A1     RESTORE REGISTERS USED FOR TIMING
660
661
662 * IF TUNED THE RCVR FOR A FINE SEGMENT.
663 * SKIP OVER FIRST COLLECTION STEP
664 * (TUNED INSTEAD F 1ST STEP)
665
666 MOVE.L A2,A0               GET RFPTR
667 SHAP L0                   GET FDFC COMMAND
668 MOVE.W #FD_MODE,L1        GET DESCRIPTOR
669 JSR RETBITS               GET MODE CODE
670 MOVE.W #1,D0              SET STEP NO. TO 1
671 JSR NQUEUE                QUEUE THE ITEMS
672 MOVE.W L0,D7              SET STEP COUNT
673 MOVE.W #TRUE,TUNED        INDICATE TUNED RCVR
674
675
676 MOVE.B IF_ASUB(A0),D0      GET SUBTYPE
677 BTST #DF_FINE,L0          CHECK IF FINE OR COARSE
678 BEQ FLOAD                 IF COARSE, GO ON
679 MOVE.W (A6)+,D0           GET NUMBER IN STEP
680 ASL.W #1,D0               ADJUST BY SIZE FOR OFFSET
681 ADD.W L0,A6               JUMP TO NEXT STEP
682
683 MOVE.W (A5)+,D0           GET NUMBER OF ACU COMMANDS
684 ASL.W #1,D0               ADJUST BY SIZE FOR OFFSET
685 ADD.W L0,A5               JUMP TO NEXT STEP
686
687 ADD.L #4,A4               JUMP TO NEXT FDFC COMMAND
688
689 BRA S FLOAD
690
691 * NO NEED TO TUNE
692 * DO 1ST COLLECTION STEP
693
694 * LOAD UP RF PROCESSOR COMMANDS
695
696 NOTUNE
697 CLR.W RFPDPOS             RESET SIO LOAD POSITION
698 MOVE.L A1,A3              GET PACKET PTR IN A3
699 MOVE.L A5,RFPTR           GET POINTER TO RFP SEQUENCE
700 JSR FFPSETUP              LOAD SIO WITH RFP COMMANDS
701 MOVE.W #SIOAINIT,D0       GET START ADDR FOR RFP CHANNEL
702 JSR CLOSEND               PREPARE SIO FOR OUTPUT
703
704

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA2.S68 NUM

```

702 * LOAD UP ACU COMMANDS
703 *
704     MOVE L A5,A0
705     JSR ACUSETUP
706     MOVE L A0,A5
707 *
708 * INITIATE 1ST COLLECTION STEP
709 *
710     MOVE L (A4)+,D0
711     JSR ADJDFC
712     JSR STARTDF
713 *
714     MOVEM.L D0/A0-A1, (SP)
715     MOVEA.L (DFA2DAT2),A0
716     MOVEA.L (DFA2OUT2),A1
717     MOVE.W (DFA2MSK2),D0
718     ZOR.W D0,(A0)
719     MOVE.W (A0),(A1)
720     MOVEM.L (SP)+,D0/A0-A1
721 *
722     SWAP D0
723     MOVE.W #FD,MODE,D1
724     JSR GETBITS
725     MOVE.W #1,D0
726     MOVE.L A6,A0
727     JSR RQUEUE
728 *
729     MOVE.L RFPFTR,A6
730     MOVE.W #0,D7
731 *
732 *****
733 *
734 * 1ST STEP STARTED, LOAD UP THE REST
735 * OF THE SEGMENT'S COMMANDS (RF PROC. AND ACU).
736 * AT THIS POINT:
737 *   A6 = RF PROCESSOR COMMAND PTR
738 *   A5 = ACU COMMAND PTR
739 *   A4 = FDF COMMAND PTR
740 *
741 *****
742 *
743 DFLDLOD
744     MOVE L A6,A0
745     MOVE.L A6,RFPFTR
746     MOVE.L A3,A6
747     MOVE.B RF_ASUB(A0),D1
748     JSR RUTPAK
749     MOVE L A0,A6
750     GET A COPY OF RFPFTR
751     PASS PTR TO RFP COMMAND LIST
752     RETURN PACKET (INPUT) PTR
753     GET SUBTYPE FOR LATER TEST
754     RETURN INPUT PACKET
755     RESTORE RFPFTR

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO, IGR, MC68000_DF] DFA2.S68 NUM

```

749  CMP W '6,D'      CHECK IF DONE YET
750  BEQ DONE        IF COUNT = TOTAL, DONE
751  *
752  MOVE W '6,D0     GET A COPY OF NO. OF STEPS
753  SUBQ W #1,D0     ALREADY DONE 1ST STEP
754  ADILP           DECR FOR COUNTER
755  SUBQ W #1,D0
756  MOVE L A5,A0     GET ACU PTR
757  *
758  * LOAD UP ALL THE REST OF THE RF PROC. COMMANDS AND
759  * ACU COMMANDS FOR THE CURRENT REGMENT
760  LOADLP          JSR RFPSETUP      LOAD ONE STEP OF RFP COMMANDS
761  JSR ACUSETUP    JSR ACUSETUP    LOAD ONE STEP OF ACU COMMANDS
762  DBF '0,LOADLF  ENDLOOP
763  *
764  MOVE L A4,FDFCOM PUT FDFC PTR INTO TEMP
765  *
766  CLR W ABRT      RESET ABORT FLAG
767  TST.W PIPE      CHECK IF PIPELINE MODE
768  BEQ WTDIF       NOT PIPE, GO WAIT
769  *
770  *
771  *
772  * DF COLLECTION LOOP
773  *
774  *
775  *
776  DFL00P          ADDQ.W #1,D7     INCREMENT COUNTER
777  CMP W '6,D7     CHECK IF DONE
778  BGT DONE
779  *
780  * INITIATE NEXT STEP
781  * (PIPELINING - DOUBLE BUFFERING)
782  *
783  STDF            MOVE.L FDFCOM,A4  GET FDFC COMMAND PTR
784  MOVE.L (A4)+,D0  GET NEXT FDFC COMMAND
785  MOVE.L A4,FDFCOM UPDATE TEMP
786  JSR ADJFDFC     ADJUST FDFC COMMAND
787  *
788  JSR STARTDF    DO IT
789  *
790  SWAP '0         GET FDFC COMMAND
791  MOVE.W #FD_MODE,D1 GET DESCRIPTOR
792  JSR GETBITS     GET MODE CODE
793  MOVE.W D7,D0    PASS STEP COUNTER
794  MOVE.L A6,A0    PASS RFPPTR
795  JSR QUEUE       QUEUE THE ITEMS

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA2.S68 NUM

```

196  •      MOVE W  (A6),D0      GET NO. PER STEP
197  •      ASL W  #1,D0        MULTIPLY BY SIZE
198  •      ADD W  D0,A6        UPDATE REPTR
199  •
200  •      TST W  PIPE          CHECK IF IN PIPELINE MODE
201  •      BNE S  WTDF         PIPELINE, GO WAIT
202  •      CMP W  #6,D7        NON PIPE, SEE IF DONE
203  •      BGE  DONE          LAST STEP IN, GO WAIT
204  •
205  •
206  •
207  •
208  •      * RECEIVING END F LOOP
209  •
210  •
211  •
212  •
213  •      JSR  WAITF          WAIT FOR DF TO COMPLETE
214  •
215  •
216  •      * A COLLECTION STEP HAS COMPLETED.
217  •      * DETERMINE WHAT TO DO WITH DATA
218  •
219  •      JSR  QUEUE         DEQUEUE ITEMS
220  •      TST W  TUNED        CHECK IF JUST TUNED RCVR
221  •      BEQ  TCAL          DID NOT TUNE, CHECK CALIB
222  •      CLR W  TUNED        CLEAR TUNED RCVR FLAG
223  •      TST W  SIGNAL       CHECK IF SIGNAL PRESENT (FINE DF)
224  •      BNE  RETDATA       GO GET E RL DATA
225  •
226  •      CLR W  D2            INITIALIZE COUNTER OF GOOD SETS
227  •      MOVE W  (A0),D0     GET NUMBER OF RAW DATA SETS
228  •      MOVE W  D0,D3       GET A COPY OF NUMBER
229  •      LEA   BUFFER,A7     GET PTR TO TEMP. BUFFER
230  •      JSR  DMPFFIFO       EMPTY DDF INTO A BUFFER
231  •
232  •      LEA   TWOHEAS(A2),A5 GET PTR TO BLA DIR. MSHRNT
233  •      JSP  DETSIG         CHECK IF SIGNAL PRESENT
234  •      TST W  D0           CHECK STATUS OF TEST
235  •      BEQ S  TAREV        NO SIG, CHECK BLA REV
236  •      MOVE W  #1,D2       SIGNAL PRES., INDICATE
237  •      *ONEHEAS A5         SET PTR TO BLA REV
238  •      JSR  DETSIG         CHECK IF SIGNAL PRESENT
239  •      TST W  D0           CHECK STATUS FLAG
240  •      BNE S  ADDNUM       IF SIGNAL PRES., UPDATE COUNTER
241  •      CLR W  D2           NOT TWO IN A ROW
242  •      BRA S  TEBL         GO CHECK RL E
243  •      ADDUM  ADDQ W  #1,D2 UPDATE COUNTER

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA2.S68 NUM

```

843 CMP.W #2,D2
844 BGE.S :GOODSIG
845
846 *
847 TERL
848 ADD.L #ONEHEAS,A5
849 JSR :ETSIG
850 ADD.W #0,D2
851 CMP.W #2,D2
852 BEQ.S :GOODSGNE
853 CMP.W #1,D2
854 BNE :BOTTOM
855 MOVE.W #TRUE,SIGFST
856 BRA :BOTTOM
857
858 *
859 GOODSIG
860 MOVE.W #TRUE,SIGNAL
861 MOVE.L A0,A4
862 MOVE.L A2,A3
863 MOVE.L ACBASE,A3
864 SUBQ.W #1,D3
865 JSR :ACCBUF
866 DBF D3,TUNEBUF
867 BRA :BOTTOM
868
869 *
870 GOODSGNE
871 MOVE.W #TRUE,SIGNAL
872 BRA :BOTTOM
873
874 *
875 TCAL
876 TST.W CALIB
877 BNE.S CALIB1
878
879 *
880 TST.W SIGNAL
881 BNE.S GETDATA
882
883 *
884 * SIGNAL NOT PRESENT.
885 * CHECK IF PRESENT, IF IT IS ACCUMULATE GOOD DATA.
886 * IF IT ISN'T, DON'T COLLECT FURTHER DATA
887
888 *
889 TST.W ABRT
890 BNE :EMPFIF
891
892 *
893 LEA :BUFFER,A2
894 MOVE.W (A0),D0
895 JSR :EMPFIFO
896
897 *
898 MOVE.L A0,A1
899 JSR :SIGPRES
900 TST.W SIGNAL
901 BNE.S ACCFB
  
```

CHECK IF FIRST TWO HAD SIGNAL
 YES, GO ACCUMULATE E BL

SET PTR TO BL E RAM DATA SET
 CHECK IF SIGNAL PRESENT
 UPDATE NUMBER BASED ON STATUS
 CHECK IF TWO IN A ROW
 SIGNAL GOOD, BUT DON'T ACC BL E
 CHECK IF ONLY BLE WAS GOOD
 NO SIGNAL, DON'T USE DATA
 ONLY BL E HAD SIGNAL.

SIGNAL PRESENT
 SET RFP COMMAND PTR
 SET DATA PTR
 GET PTR TO ACCUMULATORS
 DECR. NUMBER FOR INDEX
 UNLOAD FROM BUFFER
 ENDLOOP

SET SIGNAL PRESENT FLAG

CHECK IF CAL SEG. NO. 1
 IF YES, GO PROCESS

CHECK IF SIGNAL PRESENT
 IF YES, GO GET DATA

CHECK IF NO SIGNAL ALREADY FOUND
 UNLOAD DATA

ELSE, CHECK FOR SIGNAL PRESENCE
 GET NUMBER OF MEASUREMENTS
 UNLOAD FIFO INTO BUFFER

GET PTR TO RFP SEQUENCE
 LOOK FOR SIGNAL PRESENCE
 SEE IF SIGNAL PRESENT
 IF PRESENT, GO COLLECT

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA2.S68.NUM

```

890      MOVE.W #TRUE,ABRT      ELSE SET ABORT FLAG
891      BRA      BOTTOM
892      SUBQ.W #1,D3
893      MOVE.L ACBASE,A0      ELSE, ACCUMULATE GOOD DATA
894      JSR      ACBUF        GET PTR TO ACCUMULATORS
895      DBF      D3,BUFLOOP   UNLOAD FROM BUFFER
896      BRA.S   BOTTOM        ENDLOOP
897
898      * SIGNAL IS PRESENT, ACCUMULATE DATA
899
900      GETDATA  MOVE.L A0,A1      GET RFP COMMAND PTR
901      MOVE.L ACBASE,A0      GET PTR TO ACCUMULATORS
902      MOVE.W (A1)+,D3      GET COUNT
903      SUBQ.W #1,D3        DECR. FOR COUNTER
904      JSR      ACCUM        ACCUMULATE FROM FIFO
905      DBF      D3,FIFLOOP   ENDLOOP
906      BRA.S   BOTTOM
907
908      * CALIBRATION SEGMENT NO. 1,
909      * CHECK IF NEED TO TAKE AGC READINGS
910
911      CALIB1  CMP.W #STEPX,D0   CHECK IF 1ST READ
912      BEQ.S   FRSTREAD        IF YES GO DO IT
913      CMP.W #STEPY,D0        CHECK IF 2ND READ
914      BEQ.S   SECREAD        IF YES, GO DO IT
915      BRA.S   BOTTOM        ELSE, LOOP BACK
916
917      FRSTREAD JSR      READAGC  READ AGC VALUE OF NOISE
918      MOVE.L IPACK,A2        GET PACKET ADDR.
919      AND.W #SOOFF,D0        ONLY WANT LS BYTE
920      CMP.W NOILEV,D0        CHECK IF IN < -112 RANGE
921      BGT.S   STREAD1        GREATER THAN, GO STORE
922      MOVE.W NOILEV,D0        ELSE LESS THAN, SET TO NOISE
923      STREAD1 MOVE.B D0,SD_SSM(A2) PUT NOISE MEASURE IN
924      BRA.S   BOTTOM
925
926      SECREAD JSR      READAGC  READ AGC VALUE OF SIGNAL
927      MOVE.L IPACK,A2        GET PACKET PTR
928      MOVE.B SD_SSM(A2),D1    GET AGC VALUE OF NOISE
929      AND.W #SOOFF,D0        ONLY WANT LS BYTE
930      AND.W #SOOFF,D1        ONLY WANT LS BYTE
931      SUB.W D1,D0            CALC. SIGNAL - NOISE
932      MOVE.B D0,SD_SSM(A2)    PUT INTO PACKET
933      BRA      BOTTOM
934
935      * THROW DATA OUT (EMPTY FIFO)
936

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFA4.S68_NUM

19.3 DFA4.S68_NUM

```
***** Source Listing ==> DFA4.S68_NUM *****
*****
1  TTL      'DFA4'
2  *
3  *
4  *
5  *
6  *
7  *
8  *$INCLUDE DFA4.FMT/G
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *

I/O DEVICES USED:
DNAV      NAVCON INTERFACE

ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
ADMINUC

METHOD:

PDL:

?READW DNAV,#5,RUFADD,#0 /* READ CURRENT NAV DATA */
/* UPDATE NAV DATA IN GLOBAL DATA BASE */

NAV_LAT <- CURRENT LATITUDE
NAV_LONG <- CURRENT LONGITUDE
NAV_HEAD <- CURRENT HEADING
NAV_ROLL <- CURRENT ROLL ANGLE

?EXIT

DFA4      IDNT
          OPT      -M

XDEF      DFA4ST
XREF      NAVDATA
XREF      INAV

*$INCLUDE STRC.MAC/S
*$INCLUDE NAV.DBF/G
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA4.S68_NUM

```

44 *
45 *SINCLUDE ARTEMAS.S/G NOLIST,?READM,?EXIT
46 *
47 *SINCLUDE ARTEDATA.S/G NOLIST,RCB,DRCB
48 *
49 *      SECT      ADMINUC
50 *
51 BUFFER  DS.L  1      LATITUDE
52 LONG    DS.L  1      LONGITUDE
53 HEAD    DS.L  1      HEADING
54 PITCH   DS.L  1      PITCH ANGLE
55 ROLL     DS.L  1      ROLL ANGLE
56 *
57 BUFADD  DC.L  1      BUFFER
58 *
59 DFA4ST  ?READM DNAV,#5,BUFADD,#0 ; GET CURRENT NAV DATA
60 *
61 LEA      NAVDATA,A0    GET POINTER TO COMMON BUFFER
62 MOVE.L   BUFFER,NAV_LAT(A0) UPDATE LATITUDE
63 MOVE.L   LONG,NAV_LONG(A0) UPDATE LONGITUDE
64 *
65 * ONLY WANT MS 16 BITS OF BAM HEADING AND ROLL
66 *
67 MOVE.W   HEAD,NAV_HEAD(A0) UPDATE HEADING
68 MOVE.W   ROLL,NAV_ROLL(A0) UPDATE ROLL ANGLE
69 *
70 ?EXIT
71 *
72 END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO,IGR,MC68000_DF]DFA5.S68_NUM

19 4 DFA5.S68_NUM

```
***** Source Listing --> DFA5.S68_NUM *****
*****
1  ITL      'DFA5'
2  *
3  *
4  *
5  *
6  *
7  *
8  *SINCLUDE DFA5.RMT/G
9  *
10 *      I/O DEVICES USED:
11 *
12 *      ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
13 *      ADFUC
14 *
15 *      METHOD:
16 *
17 *
18 *
19 *
20 DFA5    IDNT
21 OPT     M,-X
22 *
23 XDEF    DFA5E
24 XREF    DFA6
25 XREF    CPUSEND
26 XREF    NAVDATA
27 XREF    GETLPK
28 XREF    HUTLPK
29 XREF    GETPAK
30 XREF    GETBITS
31 *
32 *SINCLUDE ARTEMAC S/G NOLIST,?DLINK,?EXIT,?SCHED
33 *
34 *SINCLUDE ARTEDATA S/G NOLIST,RCH
35 *
36 *SINCLUDE STRC.MAC/S
37 *
38 *SINCLUDE TFA.PAC.7
39 *
40 *SINCLUDE SD.PAC/G
41 *
42 *SINCLUDE NAV.DBS.7
43 *
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA5.S68_NUM

```

44 *INCLUDE DFR PAC 7
45 *
46 *INCLUDE DFERRS.FOU/G
47 *
48 *INCLUDE DFTYPE.FOU/G
49 *
50 *NOLIST
51 *INCLUDE MIDEQU.FOU/G
52 *LIST
53 *
54 *SECT ADFUC
55 *
56 CURFINE DS B TFA_FRI-TFA_MID 1ST PART OF TFA
57 DC B INITIALIZE FRI TO 0
58 DC B FR2 TO 0
59 DC B FR3 TO 0
60 DC B FR4 TO 0
61 DC B FR5 TO 0
62 DS B REST OF TFA
63 *
64 FINE1 EQU 1 FINE SEGMENT NO. 1
65 CALL EQU 1 CALIB SEGMENT NO. 1
66 MID EQU 1 MIDDLE SEGMENT
67 NFREQ EQU 5-1 NO. OF FREQ BYTES - 1
68 LAST EQU 1 LAST SEGMENT NO
69 DFRNIB EQU 14 ARF ID. & BLK COUNT
70 CURSEG DC W FINE1 CURRENT SEGMENT NO.
71 *
72 NBLTAB DC B 1,5,8,6 NO. OF BASELINES
73 *
74 DFA5 2DLINK A0 GET PTR TO SD PACKET
75 MOVE B 10,ASIB(A0),D0 GET SUBTYPE
76 MOVE W #D_SEG,D1 GET DESCRIPTOR
77 JSR 1,1,1 D1 < SEGMENT NO.
78 *
79 CMP B #CALIB,SP-1,1P(A0) CHECK IF CALIB
80 BNE 1,1,1 NOT CALIB
81 CMP B #CALL,D1 CHECK IF CALIB SEG NO 1
82 BNE S 1,1,1 NOT CAL. SEG NO 1
83 LEA CURFINE,A1 GET PTR TO CURRENT FINE BUFFER
84 MOVE B 10,SSM(A0),TFA_FRI(A1) GET 1ST BYTE OF FREQ
85 MOVE B 10,FR1(A0),TFA_FRI(A1) GET 1ST BYTE OF FREQ
86 MOVE B 10,FR2(A0),TFA_FRI(A1) GET 2ND BYTE OF FREQ
87 MOVE B 10,FR3(A0),TFA_FRI(A1) GET 3RD BYTE OF FREQ
88 MOVE B 10,FR4(A0),TFA_FRI(A1) GET 4TH BYTE OF FREQ
89 MOVE B 10,FR5(A0),TFA_FRI(A1) GET 5TH BYTE OF FREQ
90 MOVE W 10,ERR(A0),TFA_FRI(A1) GET ERROR CODE

```

[illegible]

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1 [ALGO IGR MC68000 DF]DFA5.S68 NUM

```

1000  M W 1,TEA,ERR,A1 COMBINE WITH ERROR TO FAR
1001  BNE
1002  MOVE W 10,ERR(A1),TEA,ERR,A1 GET INPUT ERROR CODE
1003  BNE
1004  LEA 1,URFINE,A1 GET PTR TO CURRENT DATA
1005  MOVE R 10,TEA(A1),D0 GET INPUT TYPE
1006  BNE B TEA TYPE(A1),D0
1007  RNE BADSEG DON'T MATCH, BAD SEGMENT
1008  LEA 1,TEA,ERR(A1),A1 GET PTR TO CURRENT FREQ
1009  LEA 10,ERR(A1),A1 GET PTR TO INPUT FREQ
1010  MOVE W 10,ERR(A1),D0 GET NO. OF FREQ BYTES
1011  BNE R A2,*,(A1)+ CHECK FREQ
1012  RNE BADSEG FREQ DOESN'T MATCH, BAD SEG.
1013  BNE DO_LOOP END LOOP
1014  MOVE R 10,BAND(A1),D0 GET FREQ BAND CODE
1015  EXT W 10
1016  LEA 1,NBLTAB,A1 MAKE A WORD
1017  MOVE B 1,(A2,D0),D0 GET PTR TO BL TABLE
1018  EXT W 10 GET NUMBER OF BASELINES
1019  AND W 10 CONVERT TO WORD
1020  AND W 10,00 EACH BL HAS 4 ENTRIES
1021  SUB W 1,00 DEC. FOR COUNTER
1022  LEA 10,BAND1(A0),A2 GET PTR TO INPUT DATA
1023  LEA 1,TEA,BASE(A1),A1 GET PTR TO CURRENT DATA
1024  MOVE L 1,A2,D2 GET INPUT DATA
1025  AND L 1,D2,(A1)+ ACCUMULATE WITH CURRENT
1026  DEC 10,A,CLOOP
1027  MOVE L 10,BAND1(A1),D2 GET INPUT BL E INP
1028  AND L 1,D2,TEA,BASE(A1),D0 ACCUMULATE BL E
1029  MOVE L 10,BAEQ(A1),D0 GET BL E QUAD
1030  AND L 1,D2,TEA,BAEQ(A1),D0 ACCUMULATE BL E
1031  MOVE W 10,ERR(A1),D0 GET INPUT ERROR
1032  OR W 10,TEA,ERR(A1),D0 COMBINE WITH CURRENT ERROR
1033  MOVE R 10,ASUB(A1),TEA,ASUB(A1) GET SUBTYPE
1034  OR W 10,D1 CHECK IF MIDDLE SEGMENT
1035  BNE THKLAST NOT MIDDLE
1036  LEA 1,NAV,TATA,A2 GET PTR TO NAV DATA
1037  MOVE L 1,NAV,LAT(A1),TEA,LAT(A1) GET LATITUDE
1038  MOVE L 1,NAV,LONG(A2),TEA,LON(A1) GET LONGITUDE
1039  MOVE W 1,NAV,HEAD(A2),TEA,HEAD(A1) GET HEADING
1040  MOVE W 1,NAV,ROLL(A2),TEA,ROLL(A1) GET ROLL ANGLE
1041  PLA
1042  BNE
1043  MOVE W 10,A1,D1 CHECK IF LAST SEGMENT

```

[illegible]

DF A6.S68	NUM

2

[illegible]

• THE NEW YORK PUBLIC LIBRARY •

REAR OF THE CAR - A LITTLE MORE LIGHT -
THE REAR OF THE CAR IS NOT IN THE
FRONT OF THE CAR - ANY MORE -

REAR OF THE CAR -

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

Improved GUARDRAIL V MC68000 'DF' Files
Spec 1 'ALGO' for MC68000 DF/DFA6 S68 NUM

1. The purpose of this document is to describe the
2. structure and content of the 'DF' files generated by
3. the GUARDRAIL V compiler. These files are used for
4. the execution of the compiled programs on the
5. MC68000 processor. The files are organized into
6. several sections, each containing specific data
7. and instructions. The sections are as follows:
8. a. Header section: Contains the file name, version
9. number, and other identifying information.
10. b. Data section: Contains the compiled code and
11. associated data structures.
12. c. Control section: Contains the control flow
13. information, such as branch targets and loop
14. counters.
15. d. Footer section: Contains the end of the file
16. marker and any additional comments.

17. The 'DF' files are generated by the GUARDRAIL V
18. compiler and are used for the execution of the
19. compiled programs on the MC68000 processor.

20. The structure of the 'DF' files is as follows:
21. Header section: 1-10 lines
22. Data section: 11-50 lines
23. Control section: 51-100 lines
24. Footer section: 101-110 lines

IMPROVED GUARDRAIL V MC68000 'DF' Files
 IFCD BALGO FOR MC68000 DFJDA6.S68 NUM

[illegible][illegible]

Imperial GUARDRAIL V MC68000 'DF' Files
Ref: Also for MC68000 DF/DFA6 S68 NUM

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA6.S68 NUM

327	GTACHL	TST.W	D1	CHECK IF NO SIGNAL
328		BEG.S	BLSGP	SIGNAL PRESENT
329		MOVE.W	TFA_ERR(A1),D0	GET ERROR CODE
330		BSET	#BLSGF,D0	INDICATE WARNING
331		MOVE.W	D0,DIR_FRR(A6)	UPDATE FIELD
332	BLSGF	MOVE.W	D2,BADBL	SET BAD BASELINE
333	*	MOVE.B	TFA_ASUB(A5),D0	GET SUBTYPE
334		BTEST	#TFA_FINE,D0	CHECK IF FINE OR COARSE
335		BNE.S	FINENAV	IF FINE, XFER NAV DATA
336		LEA	NAVDATA,A1	GET PTR TO DATA BASE
337		MOVE.L	NAV_LAT(A1),DFR_LAT(A6)	GET LATITUDE
338		MOVE.L	NAV_LONG(A1),DFR_LONG(A6)	GET LONGITUDE
339		MOVE.W	NAV_HEAD(A1),DFR_HEA(A6)	HEADING
340		MOVE.W	NAV_ROLL(A1),DFR_ROLL(A6)	ROLL ANGLE
341		BRA.S	TSTROLL	GO TEST ROLL ANGLE
342	*			
343	FINENAV	MOVE.L	TFA_LAT(A5),DFR_LAT(A6)	XFER LATITUDE
344		MOVE.L	TFA_LONG(A5),DFR_LONG(A6)	XFER LONGITUDE
345		MOVE.W	TFA_HEA(A5),DFR_HEA(A6)	HEADING
346		MOVE.W	TFA_ROLL(A5),DFR_ROLL(A6)	ROLL ANGLE
347	*			
348	TSTROLL	MOVE.W	INTF_ROLL(A6),D0	GET ROLL ANGLE
349		BPL.S	POSROLL	IF ROLL IS POSITIVE, OKAY
350		NEG.W	D0	ELSE CONVERT TO POSITIVE
351		CMP.W	MAXROLL,D0	CHECK IF TOO LARGE
352	POSROLL	BLE.S	GTNUM	IF LESS, THEN OKAY
353		MOVE.W	DFR_ERR(A6),D0	GET ERROR CODE
354		BSET	#ROLLANG,D0	INDICATE ROLL ANGLE
355		MOVE.W	D0,DFR_ERR(A6)	UPDATE FIELD
356	*			
357	GTNUM	MOVE.B	TFA_BAND(A5),D0	GET FREQ. BAND CODE
358		EXT.W	D0	MAKE A WORD
359		MOVE.W	D0,BANDLB	SAVE BAND CODE
360		LEA	NTAB,A1	GET PTR TO TABLE OF NUM.S
361		MOVE.B	D(A1,D0.W),D5	GET NUMBER OF BASELINES
362		EXT.W	D5	MAKE A WORD
363		MOVE.W	D5,NBL	SAVE NUMBER
364	*			
365		LEA	PIFFER,A1	GET PTR TO TEMP BUFFER
366		MOVE.L	A5,A0	GET PTR TO INPUT PACKET
367		JSR	14SQUAD	CONVERT TO DIR & REV. ANGLES
368	*			
369		LEA	DFR_BASE(A6),A2	GET PTR TO FINAL BUFFER
370		MOVE.L	A2,THETA	SAVE PTR TO FINAL ANGLES
371		JSR	DIRREV	COMBINE DIR., REV. RCVR PHASE

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFA6.S68_NUM

```

373 MOVE.W DFR_ERR(A6),D0      GET ERROR CODE SO FAR
374 AND.W #ERRM:K,D0          CHECK FOR FATAL ERRORS
375 BNE      ENDRESP           IF PRESENT, GO SEND IT
376 *
377 LEA      TFA_FR1(A5),A0     GET PTR TO HCD FREQ
378 MOVE.W #NFREQ,D0           GET NUMBER OF DIGITS
379 JSR      BCDBIN             CONVERT TO INTEGER FREQ.
380 MOVE.L D1,INFRQ            SAVE INT. FREQ.
381 *
382 TST.B ADFBIT               CHECK IF ADF BITE
383 BNE      ADFBITP            GO PROCESS ADF BITE (NO LOB)
384 *
385 *-----*
386 MOVEA.L (DFA6DAT3),A3      ;ADDR OF TIMING WORD #3
387 MOVEA.L (DFA6OUT3),A4      ;ADDR TO OUTPUT TIMING WORD #3
388 MOVE.W (DFA6MSK3),D6       ;MASK USED TO SET TIMING BIT #3
389 OR.W D6,(A3)               ;SET TIMING BIT #3
390 MOVE.W (A3),(A4)           ;OUTPUT TIMING WORD #3
391 *-----*
392 *
393 JSR      COARSE             PERFORM 1ST PASS SEARCH
394 *
395 *-----*
396 EOR.W D6,(A3)              ;CLEAR TIMING BIT #3
397 MOVE.W (A3),(A4)           ;OUTPUT TIMING WORD #3
398 *-----*
399 *
400 *
401 TST.B NORES                CHECK STATUS
402 BEQ.S SECSRCH              NO ERRORS, GO TO 2ND SEARCH
403 MOVE.B NORES,D1            GET ERROR BIT NUMBER
404 EXT.W D1                    MAKE A WORD
405 MOVE.W DFR_ERR(A6),D0      GET ERRORS SO FAR
406 BSET D1,D0                 INDICATE ERROR IN 1ST PASS
407 MOVE.W D0,DFR_ERR(A6)      UPDATE FIELD
408 BRA      ENDRESP
409 *
410 SECSRCH LEA      TFA_FR1(A5),A0     GET PTR TO FREQ.
411 MOVE.W #N10KHZ,D0          ONLY WANT TO 10 KHZ
412 JSR      BCDBIN             CONVERT TO BINARY
413 ADD.L #RNDFRQ,D1           ROUND TO MHZ
414 DIVS #KHZ,D1               CONVERT TO MHZ
415 MOVE.W D1,IFRQMHZ          SAVE FREQ. IN MHZ
416 *
417 JSR      QTHRESH            DETERMINE QTHRESH
418 *
419 MOVE.W DFR_ROLL(A6),ROLL GET ROLL ANGLE

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFA6.S68_NUM

```

420 * -----
421 * MOVEA.L (DFA6DAT4),A3 ;ADDR OF TIMING WORD #4
422 * MOVEA.L (DFA6OUT4),A4 ;ADDR TO OUTPUT TIMING WORD #4
423 * MOVE.W (DFA6MSK4),D6 ;MASK USED TO SET TIMING BIT #4
424 * OR.W D6,(A3) ;SET TIMING BIT #4
425 * MOVE.W (A3),(A4) ;OUTPUT TIMING WORD #4
426 * -----
427 *
428 * JSR P2SRCH PERFORM 2ND PASS SEARCH
429 *
430 * -----
431 * EOR.W D6,(A3) ;CLEAR TIMING BIT #4
432 * MOVE.W (A3),(A4) ;OUTPUT TIMING WORD #4
433 * -----
434 *
435 * TST.B MORES2 CHECK STATUS
436 * BEQ GOODLOB IF 0 THEN NO ERRORS
437 * MOVE.W DFR_ERR(A6),D0 GET ERRORS SO FAR
438 * RSET #PFIT,D0 INDICATE ERROR
439 * MOVE.W D0,DFR_ERR(A6) UPDATE FIELD
440 *
441 * GOODLOB MOVE.W LOB_DFR_LOB(A6) RETURN LOB
442 * MOVE.L DUAL,D0 GET QUALITY
443 * BFL.S FNDMAG CHECK IF NEGATIVE
444 * NEG.L D0 ALWAYS WANT POSITIVE
445 * CLR.W D1 CLEAR MAGNITUDE COUNT
446 *
447 * MAGLOOP TST.L D0 CHECK IF 0 YET
448 * BEQ.S RETQUAL IF 0 THEN DONE
449 * ASR.L #1,D0 ADJUST VALUE
450 * ADDQ.W #1,D1 INCREMENT MAGNITUDE COUNT
451 * BRA.S MAGLOOP ENDLOOP
452 *
453 * RETQUAL MOVE.W D1,DFR_QUAL(A6) RETURN QUALITY (MAGNITUDE)
454 * BRA ENDRESP
455 * -----
456 *
457 * * ARE DF BITF PROCESSING
458 *
459 * -----
460 * ADFBITP CLR.W D4 INITIALIZE RESULTS (ALL FAIL)
461 * LEA BITETBL,A0 GET PTR TO EXPECTED VALUES
462 * CLR.W D0 INITIALIZE INDEX
463 * MOVE.W #NBITE,D1 GET NUMBER OF FREQ.S
464 * MOVE.L D1,DFR_FREQ,D2 GET INTEGER DF FREQUENCY
465 *
466 * RETBITP CMF.L A0)+,D2 CHECK IF DF FREQ.

```


Improved GUAPDRAIL V M068000 'DF' Files
DECI |ALGO FOR M068000 DF|DFA6 S68 NPM

608	FOR W	CALL	TRANSITION FILE
609	MOV W	ALL AN	TRANSITION FILE
610			
611		PERIT	
612		END	

[illegible]

[illegible]

Improved GUARDRAIL V MC68000 D1 [DRAFT] 1984

```

185      JCR      PUSH
186      CMP.W    #TBITE,D0
187      BNE.S    EXIT
188      MOVE.B   DG_ASUB(A3),D2
189      BTST     #DG_FINE,D2
190      BNE.S    EXIT
191      CMP.LB   #DGMID,DG_MID(A3)
192      ; and not in PE request msg (to
193      ; protect against link problems)
194      not DG msg, therefore not from
195      ; ARDFBIT routine...exit
196      BNE.C    EXIT
197      MOVE.L   DG_LINK(A3),A4
198      CMP.L    #NULL,A4
199      BEQ.S    EXIT
200      MOVE.L   A3,A6
201      JCR      PUTPAK
202      MOVE.L   A4,A3
203      BRA      TOP
204
205      MOVE.L   A3,A6
206      JCR      PUTPAK
207
208      ; NO LINK D1
209
210      ; CHECK NO. OF REQUESTS PREVIOUSLY
211      ; IF >0 THEN DONE
212      ; START CHECKING FOR OLD RQSTS

```

AD-A184 057 US ARMY INTELLIGENCE CENTER AND SCHOOL (USAICS)
SOFTWARE ANALYSIS AND MAN (U) JET PROPULSION LAB
PASADENA CA 8 PARDO 05 MAR 87 JPL-D-4216 NAS7-918

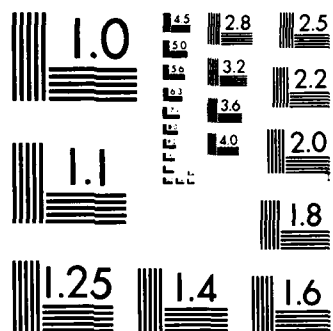
AD-A184 057 US ARMY INTELLIGENCE CENTER AND SCHOOL (USAICS)
SOFTWARE ANALYSIS AND MAN (U) JET PROPULSION LAB
PASADENA CA 8 PARDO 05 MAR 87 JPL-D-4216 NAS7-918

AD-A184 057 US ARMY INTELLIGENCE CENTER AND SCHOOL (USAICS) 5/5
SOFTWARE ANALYSIS AND MAN (U) JET PROPULSION LAB
PASADENA CA 8 PARDO 05 MAR 87 JPL-D-4216 NAS7-918

UNCLASSIFIED F/G 12/5

UNCLASSIFIED F/G 12/5

UNCLASSIFIED F/G 12/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG2.S68_NUM

19.8 DFG2.S68_NUM

```

***** Source Listing --> DFG2.S68_NUM *****
*****
1  TTL      'DFG2 - REMOVE OLD DF REQUESTS'
2
3  -----
4
5  VERSION  DATE
6  1.0      5/11/83
7
8  *$INCLUDE DFG2.FMT/G
9
10 I/O DEVICES USED: NONE
11
12 ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
13 GADMINUC
14
15 METHOD:
16 FOR EACH TYPE OF DF (EACH DF QUEUE) THE HEADER
17 BLOCK IS PASSED TO FINDOLD. FINDOLD DETERMINES
18 THE LIST ENTRIES THAT HAVE BEEN ON THE QUEUE
19 TOO LONG, AND REMOVES THEM. IF, AFTER SEARCHING
20 EACH DF REQUEST QUEUE, THERE ARE NO ENTRIES LEFT,
21 THIS TASK IS NOT RESCHEDULED. OTHERWISE, THIS
22 TASK RESCHEDULES ITSELF TO CHECK THE LISTS AGAIN.
23
24 * PDL:
25
26 SUM <= 0      /* INITIALIZE SUM */
27 * ?LOCKW HBLOK
28 * LOOP FOR EVERY HEADER BLOCK IN DF TYPE TABLE (EXCEPT GEO)
29 * CALL FINDOLD(HDRBLK) /* SCREEN QUEUE */
30 * ENDLOOP
31 * ?UNLOC HBLOK
32
33 IF SUM > 0      /* CHECK IF ANY REQUESTS LEFT */
34 THEN           /* NON-EMPTY QUEUES */
35   ?SCHED DFG2,#20,#-1
36 * ENDIF
37
38 * ?EXIT
39
40
41 -----
42
43 DFG2  IDNT

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG2.S68_NUM

44	OPT	-M	
45	*		
46	XDEF	DFG2E,SUM	
47	XREF	DFG2	
48	XREF	DFTYPTAB	
49	XREF	HBLOK	
50	XREF	FINDOLD	
51	*		
52	*\$INCLUDE	ARTEMACS.S/G NOLIST,?LOCKW,?UNLOC,?SCHED,?EXIT	
53	*		
54	*\$INCLUDE	ARTEDATA.S/G NOLIST,TIB,RIB	
55	*		
56	*\$INCLUDE	NDF.EQU/G	
57	*		
58	SECT	GADMINUC	
59	*		
60	SUM	DS.W 1	
61	*		
62	DFG2E	CLR.W SUM	INITIALIZE TOTAL NO. OF ENTRIES
63	*	?LOCKW HBLOK	
64	*		
65	MOVE.W	#NDF,D0	GET NUMBER OF DF TYPES
66	SUBQ.W	#1,D0	AOI AND GEO ARE SAME LIST
67	SUBQ.W	#1,D0	DECR. FOR COUNTER
68	LEA	DFTYPTAB,A1	GET ADDRESS OF HDR BLK TABLE
69	*		
70	LOOP	MOVE.L (A1)+,A0	GET HDR BLK POINTER
71	JSR	FINDOLD	CALL FINDOLD
72	DBF	D0,LOOP	ENDLOOP
73	*		
74	?UNLOC	HBLOK	
75	TST.W	SUM	CHECK TOTAL NUMBER
76	BEQ.S	DONE	LISTS NOW EMPTY
77	*		
78	?SCHED	DFG2,#4,#20,#-1	
79	*		
80	DONE	?EXIT	
81	END		

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

19.9 DFG3.S68_NUM

***** Source Listing ==> DFG3.S68_NUM *****

```

1  TTL      'DFG3'
2  *
3  *-----*
4  *
5  *
6  *   VERSION   DATE
7  *   1.0      5/26/83
8  *   1.1      1/16/84 PYW - INSERT TIMING BIT
9  *   1.2      4/27/84
10 *   1.3      5/8/84
11 *   1.4      7/20/84 PYW - Force Calibration DF to send ARF
12 *                                     id of 1 only....due to no PE
13 *                                     specification and design of doing
14 *                                     cal on one plane at a time
15 *   1.5      8/20/84 PKZ - DISABLE OR ENABLE DF REPORTING
16 *                                     (SENDING OF UDR1,UDR2,UDR3) BASED
17 *                                     ON TFLAG1 AND TFLAG2 (IN DFRPT).
18 *   1.6      8/29/84 PYW - INSERT PIO BIT TO INDICATE COARSE
19 *                                     OR FINE DF
20 *$INCLUDE DFG3.FMT/G
21 *
22 *   I/O DEVICES USED: NONE
23 *
24 *   ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
25 *   GADMINUC
26 *
27 *   METHOD:
28 *
29 *   PDL:
30 *
31 *   /* VARIABLE DEFINITION :
32 *   PROCESSING A FINE = FLAG, TRUE/FALSE (FINE/COARSE)
33 *   PROCESSING ADF BITE = FLAG, TRUE/FALSE
34 *   PROCESSING A CALIB = FLAG, TRUE/FALSE
35 *   CURRENT DF = DF MESSAGE TO BE SENT UP THE LINK
36 *   CURRENT FINE DF = CURRENT FINE DF REQUEST BEING PROCESSED
37 *   SEG NO. = SEGMENT NUMBER, PART OF CURRENT FINE DF
38 *   CONTIG = CONTINGUOUS FLAG, PART OF CURRENT FINE DF
39 *   RESPOND = SEND RESPONSE FLAG, PART OF CUR. FINE DF
40 *   TUNE REQ. = TUNE ACORR RCVR FLAG
41 *   DFIDLE = FLAG, TRUE/FALSE
42 *   AUDIO CORR REQ. = FLAG, YES/NO
43 *   ACORR. AVAILABLE = FLAG, TRUE/FALSE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000_DF]DFG3.S68_NUM

```

44 * CUR. FINE ACORR      - FLAG, TRUE/FALSE  CURRENT FINE HAS
45 *                      REQUESTED AND BEEN GIVEN AUDIO CORR.
46 * SIGNAL GEN. REQ.    - FLAG, YES/NO
47 * HBLOK               - DF REQUEST HEADER BLOCK DATA BASE
48 * LAST SEGMENT        - FLAG, COARSE/FINE
49 * INTEROP             - FLAG, TRUE/FALSE (PROCESSING AN INTEROP)
50 *
51 * IF INTEROP=TRUE     /* CHECK IF PROCESSING AN INTEROP REQUEST */
52 * THEN                /* DON'T WANT TO SEND THE NEXT REQUEST */
53 * ?EXIT
54 * ENDF
55 *
56 * IF PROCESSING A FINE
57 * THEN                /* CHECK IF DOING ARF DF BITE */
58 * IF PROCESSING ADF BITE
59 * THEN                /* GET NEXT ARF DF BITE REQUEST */
60 * ?LOCKW HBLOK        /* LOCK DATA BASE */
61 * CALL GETFINE(PTR,FLAG) /* BITE HIGHEST PRIORITY */
62 * CURRENT DF <= DF REQUEST PARAMETERS /* MOVE PARAMS. */
63 * ?UNLOC HBLOK        /* FREE DATA BASE */
64 * IF LAST ADF BITE REQUEST
65 * THEN                /* RESET FLAGS */
66 * PROCESSING A FINE <= FALSE
67 * PROCESSING ADF BITE <= FALSE
68 * ENDF
69 * ELSE                /* CHECK IF DOING A CALIBRATION */
70 * IF PROCESSING A CALIB
71 * THEN                /* DO THE NEXT SEGMENT */
72 * SEG NO. <= SEG NO. - 1 /* COUNTS DOWN */
73 * IF 2ND CALIB SEGMENT /* CHECK FOR CONTIG. */
74 * THEN                /* NOT CONTIGUOUS */
75 * CONTIG. <= NO
76 * ELSE                /* 3RD THROUGH LAST SEGMENT */
77 * CONTIG. <= YES /* ALWAYS CONTIGUOUS */
78 * ENDF
79 * IF LAST SEGMENT OF CALIB
80 * THEN                /* TAKE CARE OF FLAGS */
81 * RESPOND <= YES
82 * PROCESSING A CALIB <= FALSE
83 * PROCESSING A FINE <= FALSE
84 * ENDF
85 * CURRENT DF <= CURRENT FINE DF /* XFER PARAMS. */
86 * ELSE /* NOT PROC. A CALIB, OKAY TO INTERLEAVE */
87 * IF LAST SEGMENT = FINE
88 * THEN                /* TRY TO GET A COARSE */
89 * ?LOCKW HBLOK        /* LOCK DATA BASE */
90 * CALL GETCRS(PTR,FLAG) /* CHECK IF ANY REQUESTS */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *

IF NO MORE COARSE DF'S
  THEN /* CONTINUE WITH FINE */
  ?UNLOC HBLOCK /* DON'T NEED LISTS */
  SEG NO. <= SEG NO. - 1
  CONTIG <= YES
  IF LAST SEGMENT OF FINE
    THEN /* CLEAN UP FLAGS */
    RESPOND <= YES
    PROCESSING A FINE <= FALSE
  ENDIF
  CURRENT DF <= CURRENT FINE DF
  ELSE /* USE THE COARSE */
  CURRENT DF <= DF PARAMS /* FROM PTR */
  ?UNLOC HBLOCK /* FREE DATA BASE */
  LAST SEGMENT <= COARSE
  ENDIF
  ELSE /* LAST SEG-COARSE, CONT. WITH FINE */
  SEG NO. <= SEG NO. - 1
  IF LAST SEGMENT OF FINE
    THEN /* CLEAN UP FLAGS */
    RESPOND <= YES
    PROCESSING A FINE <= FALSE
  ENDIF
  CURRENT DF <= CURRENT FINE DF
  LAST SEGMENT <= FINE
  ENDIF
  ENDIF /* PROCESSING A CALIB */
  ENDIF /* PROCESSING ADF BITE */

  ELSE /* NOT PROCESSING A FINE */
  ?LOCKM HBLOCK
  CALL GETFINE(PTR, FLAG) /* TRY TO GET A FINE */
  IF NO MORE FINE REQUESTS /* STATUS FLAG */
    THEN /* TRY TO GET A COARSE */
    CALL GETCRS(PTR, FLAG)
    IF NO MORE COARSE REQUESTS /* STATUS FLAG */
      THEN /* CHECK FOR CALIB */
      CALL GETDF(CALIB) /* LOOK FOR A CALIB REQUEST */
      IF NO MORE CALIB REQUESTS /* STATUS FLAG */
        THEN /* NOTHING TO DO */
        ?UNLOC HBLOCK /* FREE DATA BASE */
        DFIDLE <= TRUE
        ELSE /* USE CALIB REQUEST */
        GO TO USEFINE
      ENDIF
    ELSE /* USE THE COARSE DF */
    CURRENT DF <= DF PARAMETERS /* FROM PTR */
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

138 *      ?UNLOC HBLOCK /* FREE DATA BASE */
139 *      DFIDLE <= FALSE
140 *
141 *      ENDIF
142 *      ELSE /* USE THE FINE REQUEST */
143 *      USEFINE.CURRENT FINE DF <= DF PARAMETERS /* FROM PTR */
144 *      ?UNLOC HBLOCK /* FREE DATA BASE */
145 *      CURRENT DF <= CURRENT FINE DF /* CREATE REQUEST MESSAGE */
146 *      DFIDLE <= FALSE
147 *      IF TYPE=INTEROP
148 *      THEN /* SET APPROPRIATE FLAGS */
149 *      INTEROP <= TRUE
150 *      PROCESSING A FINE <= FALSE /* INTEROP LIKE COARSE */
151 *      ELSE /* NOT INTEROP */
152 *      IF TYPE=ADF BITE
153 *      THEN /* SET APPROPRIATE FLAGS */
154 *      PROCESSING ADF BITE <= TRUE
155 *      PROCESSING A FINE <= TRUE
156 *      ELSE /* NOT PROCESSING ADF BITE */
157 *      IF TYPE=BITE /* CHECK FOR SYSTEM BITE */
158 *      THEN /* SET FLAGS, REQUEST SIGNAL GEN. */
159 *      SIGNAL GEN. REQ. <= YES
160 *      PROCESSING A FINE <= TRUE
161 *      LAST SEGMENT <= FINE
162 *      ELSE
163 *      IF TYPE=CALIB
164 *      THEN /* SET FLAGS AND REQUEST SIGNAL GEN. */
165 *      PROCESSING A CALIB <= TRUE
166 *      PROCESSING A FINE <= TRUE
167 *      SIGNAL GEN. REQ. <= YES
168 *      ELSE /* NOT A CALIB. */
169 *      CUR FINE ACORR <= FALSE /* ASSUME NOT */
170 *      IF AUDIO CORR. REQUESTED
171 *      THEN /* CHECK IF AUDIO CORR. AVAILABLE */
172 *      IF ACORR AVAILABLE
173 *      THEN /* REQUEST */
174 *      AUDIO CORR REQ. <= YES
175 *      CUR FINE ACORR <= TRUE
176 *      ACORR AVAILABLE <= FALSE
177 *      ELSE /* UNAVAILABLE */
178 *      AUDIO CORR REQ. <= NO
179 *      TUNE REQ. <= FALSE
180 *      ENDIF
181 *      ENDIF /* ACORR REQUESTED */
182 *      PROCESSING A FINE <= TRUE
183 *      LAST SEGMENT <= FINE
184 *      ENDIF /* TYPE = CALIB */

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

185 *          ENDIF /* TYPE = SYSTEM BITE */
186 *          ENDIF /* TYPE = ADF BITE */
187 *          ENDIF /* TYPE = INTEROP */
188 *          ENDIF /* NO MORE FINE REQUESTS */
189 *          ENDIF /* PROCESSING A FINE */
190 *
191 * IF NOT(DFIDLE)
192 *   THEN /* SEND THE DF REQUEST */
193 *     CALL GETPAK(IPACK) /* GET A PACKET */
194 *     IPACK <= CURRENT DF /* FORMAT MESSAGE */
195 *     IF RESPOND = YES
196 *       THEN /* NEED A CONTROL PACKET AS WELL */
197 *         CALL GETPAK(ICTL) /* GET ANOTHER PACKET */
198 *         FORMAT CONTROL PACKET
199 *         ICTL <= IPACK /* LINK MESSAGE TO CONTROL PACKET */
200 *         ICTL <= ACCOUNTABILITY OF DF REQUEST
201 *         BASED ON TYPE, GET RETURN ADDRESS /* TABLE */
202 *         ICTL <= RETURN ADDRESS /* PUT INTO CONTROL PACKET */
203 *         CALL GASEND(ICTL) /* SEND CONTROL PACKET */
204 *         ELSE /* JUST SEND THE DF REQUEST */
205 *           CALL GASEND(ICTL) /* SEND PACKET */
206 *         ENDIF
207 *       ENDIF
208 *
209 *
210 *
211 *
212 * DFG3
213 *   IDNT
214 *   OPT -M,-X
215 *   DFG3
216 *   XDEF ACORREQ
217 *   XDEF ACORAV
218 *   XDEF SIGREQ
219 *   XDEF INTEROP
220 *   XDEF CURDF
221 *   XDEF GETBITS
222 *   XDEF PUTBITS
223 *   XDEF GETFINE
224 *   XDEF GETCRS
225 *   XDEF GETPAK
226 *   XDEF GETDF
227 *   XDEF DFPEZ
228 *   XDEF ARDFBTR
229 *   XDEF ADFCOL
230 *   XDEF GSDF
231 *   XDEF ACIDF
232 *   XDEF GEOLCOL

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

232 XREF GASEND
233 XREF HBLOK
234 XREF DFTYPTAB
235 XREF DFG3DAT1
236 XREF DFG3OUT1
237 XREF DFG3MSK1
238 XREF DFG3DAT2
239 XREF DFG3OUT2
240 XREF DFG3MSK2
241
242 XREF TFLAG1,TFLAG2
243 XREF DPMODE
244
245 *$INCLUDE STRC.MAC/S
246 *
247 *$INCLUDE ARTEMACS.S/G NOLIST,?LOCKH,?UNLOC,?EXIT
248 *
249 *$INCLUDE ARTEDATA.S/G NOLIST,TIB
250 *
251 *$INCLUDE DF1.DBS/G
252 *
253 *$INCLUDE DF.PAC/G
254 *
255 *$INCLUDE CTL.PAC/G
256 *
257 *$INCLUDE DFTYPE.EQU/G
258 *
259 NOLIST
260 *$INCLUDE MIDEQU.EQU/G
261 *
262 LIST
263 MACRO AS,AD
264 MOVE.W DF1_TIM(AS),DF1_TIM(AD)
265 MOVE.B DF1_MID(AS),DF1_MID(AD)
266 MOVE.W DF1_ACC(AS),DF1_ACC(AD)
267 MOVE.B DF1_TYP(AS),DF1_TYP(AD)
268 MOVE.B DF1_ASUB(AS),DF1_ASUB(AD)
269 MOVE.B DF1_FR1(AS),DF1_FR1(AD)
270 MOVE.B DF1_FR2(AS),DF1_FR2(AD)
271 MOVE.B DF1_FR3(AS),DF1_FR3(AD)
272 MOVE.B DF1_FR4(AS),DF1_FR4(AD)
273 MOVE.B DF1_FR5(AS),DF1_FR5(AD)
274 MOVE.B DF1_GAIN(AS),DF1_GAIN(AD)
275 MOVE.B DF1_RNG(AS),DF1_RNG(AD)
276 MOVE.B DF1_FLAG(AS),DF1_FLAG(AD)
277 MOVE.B DF1_FOR(AS),DF1_FOR(AD)
278 MOVE.B DF1_DSE(AS),DF1_DSE(AD)

```

```

;addr of timing word
;addr to output timing word to
;mask used to set/clear timing bit
;addr of timing word for bit #2
;addr to output timing word to
;mask to indicate coarse (cleared)
; or fine (set) DF

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

279 MOVE.L DF1_SIG(AS),DF1_SIG(AD)
280 MOVE.B DF1_ARF(AS),DF1_ARF(AD)
281 MOVE.B DF1_BND(AS),DF1_BND(AD)
282 MOVE.B DF1_ADFC(AS),DF1_ADFC(AD)
283 MOVE.B DF1_ARTF(AS),DF1_ARTF(AD)
284 ENDM
285 *
286 MACRO AS,AD
287   MOVE.B #DFMID,DF_MID(AD)
288   MOVE.B #DFNIB,DF_NIB(AD)
289   MOVE.W DF1_ACC(AS),DF_ACC(AD)
290   MOVE.B DF1_FR1(AS),DF_FR1(AD)
291   MOVE.B DF1_FR2(AS),DF_FR2(AD)
292   MOVE.B DF1_FR3(AS),DF_FR3(AD)
293   MOVE.B DF1_FR4(AS),DF_FR4(AD)
294   MOVE.B DF1_FR5(AS),DF_FR5(AD)
295   MOVE.B DF1_GAIN(AS),DF_GAIN(AD)
296   MOVE.B DF1_RNG(AS),DF_RNG(AD)
297   MOVE.B DF1_FLAG(AS),DF_FLAGS(AD)
298   MOVE.B DF1_TYP(AS),DF_TYP(AD)
299   MOVE.B DF1_ASUB(AS),DF_ASUB(AD)
300 ENDM
301 *
302 *
303 *
304 *
305 DS.W 0
306 CURDF DS.B DF1
307 DS.W 0
308 CURFINE DS.B DF1
309 *
310 TRUE EQU 1
311 FALSE EQU 0
312 YES EQU 1
313 NO EQU 0
314 FINE EQU 1
315 COARSE EQU 0
316 COARTIME EQU 2
317 PACHEAD EQU -4
318 *
319 PFINE DC.B FALSE
320 PABITE DC.B FALSE
321 PCALIB DC.B FALSE
322 DFIDLE DC.B FALSE
323 ACORREQ DC.B NO
324 ACORAV DC.B TRUE
325 SIGREQ DC.B NO

```

(1/2 SECOND TIMEOUT FOR COARSE DF)
 PACKET HEADER OFFSET
 PROCESSING A FINE FLAG
 PROCESSING ADF BITE FLAG
 PROCESSING A CALIB FLAG
 DF IDLE FLAG
 AUDIO COR. REQUEST
 AUDIO COR. AVAILABLE
 SIGNAL GENERATOR REQUEST

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG3.S68_NUM

326	INTEROP	DC.B	FALSE	INTEROP DF FLAG
327	LASTSEG	DC.B	COARSE	LAST SEGMENT
328	CFACR	DC.B	FALSE	CURRENT FINE HAS ACORR.
329	*			
330	LAST	EQU	1	LAST SEGMENT COUNTER
331	LASTADF	EQU	3	LAST ADF BITE SEGMENT NO.
332	SCNDAL	EQU	5	2ND CALIB SEGMENT NO.
333	FRSTCAL	EQU	6	1ST CALIB SEGMENT NO.
334	TIMEOUT	EQU	2	TIMEOUT AMOUNT
335	*			
336	DFNIB	EQU	S32	ARF ID. AND BLOCK COUNT
337	*			
338	BITPRI	DC.B	4	NORMAL BITE RESPONSE
339	ABITPRI	DC.B	5	ARF DF BITE RESP.
340	RESPRI	DC.B	4	DFPE2
341		DC.B	4	DFPE2 (INTEROP)
342		DC.B	4	ADFCOL
343		DC.B	4	GS DF
344		DC.B	4	DFPE2 (CALIB)
345		DC.B	4	AOI
346		DC.B	4	GEOSCREEN
347	*			
348	BITRES	DC.L	DFPE2	NORMAL BITE
349	ABITRES	DC.L	ARDFBITR	ARF DF BITE
350	RETAB	DC.L	DFPE2	MANUAL
351		DC.L	DFPE2	INTEROP
352		DC.L	ADFCOL	AUTO DF
353		DC.L	GSDF	GS DF
354		DC.L	DFPE2	CALIB
355		DC.L	AOIDF	AOI
356		DC.L	GEOCOL	GEOSCREEN
357	*			
358	DFG3		MOVEM.L A0-A6/D0-D7, -(SP)	SAVE REGISTERS
359			MOVEA.L (DFG3DAT1), A0	ADDR OF TIMING WORD
360			MOVEA.L (DFG3OUT1), A1	ADDR TO OUTPUT TIMING WORD TO
361			MOVE.W (DFG3MSK1), D0	MASK USED TO SET TIMING BIT
362			OR.W D0, (A0)	SET TIMING BIT
363			MOVE.W (A0), (A1)	OUTPUT TIMING WORD
364			TST.B INTEROP	CHECK IF PROC. AN INTEROP
365			BNE DONE	IF INTEROP, EXIT
366	*			
367			TST.B PFINE	CHECK IF PROC. A FINE
368			BEQ NOTFINE	NOT PROC. A FINE
369	*			
370			* PROCESSING A FINE, CHECK IF PROC. ADF BITE	
371	*			
372			TST.B PABITE	CHECK IF PROC. ADF BITE

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

373      BEQ      NOTABITE      NOT PROC. ADF BITE
374      *
375      * PROCESSING ADF BITE, GET NEXT REQUEST
376      *
377      ?LOCKM  HBLOK
378      JSR      GETFINE
379      LEA      CURDF,A0
380      DFPARM  A1,A0
381      ?UNLOC  HBLOK,A0
382      MOVE.B  DF1,ASUB(A0),D0
383      MOVE.W  #DF_SEG,D1
384      JSR      GETBITS
385      CMP.W   #LASTADF,D1
386      BNE     SENDDF
387      MOVE.B  #FALSE,PFINE
388      MOVE.B  #FALSE,PAFITE
389      BRA     SENDDF
390      *
391      * NOT PROCESSING ADF BITE, CHECK IF PROC. A CALIB
392      *
393      NOTABITE 1ST.B  PCALIB      CHECK IF PROC. A CALIB
394      BEQ      NOTCALIB      NOT PROC. A CALIB
395      *
396      * PROCESSING A CALIB, GET NEXT SEGMENT
397      *
398      LEA      CURFINE,A0
399      MOVE.B  DF1,ASUB(A0),D0
400      MOVE.W  #DF_SEG,D1
401      JSR      GETBITS
402      SUBQ.W  #1,D1
403      MOVE.W  D1,D2
404      MOVE.W  #DF_SEG,D1
405      JSR      FUTBITS
406      CMP.W   #SCNDCAL,D2
407      BNE.S   CONTCAL
408      BCLR    #DF_CONT,D0
409      BRA.S   LTCAL1ST
410      CONTCAL BSET    #DF_CONT,D0
411      LTCAL1ST CMP.W  #LAST,D2
412      BNE.S   NTL1STCAL
413      BSET    #DF_RESP,D0
414      MOVE.B  #FALSE,PCALIB
415      MOVE.B  #FALSE,PFINE
416      NTL1STCAL MOVE.B  D0,DF1,ASUB(A0)
417      LEA      CURDF,A1
418      DFPARM  A0,A1
419      BRA     SENDDF

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

420 *
421 * NOT PROCESSING ADF BYTE OR CALIB,
422 * OKAY TO INTERLEAVE
423 *
424 NOTCALIB TST.B LASTSEG      CHECK WHAT LAST SEGMENT WAS
425 BEQ COARS                  LAST SEGMENT WAS A COARSE
426 *
427 * LAST SEGMENT WAS A FINE,
428 * TRY TO GET A COARSE REQUEST
429 *
430 ?LOCKW HBLOK
431 JSR GETCRS                  TRY TO GET A COARSE DF
432 TST.W D1                   CHECK STATUS FLAG
433 BNE USECRS                  FOUND A COARSE REQUEST
434 *
435 * NO COARSE FOUND, USE NEXT SEGMENT OF FINE
436 *
437 ?UNLOC HBLOK
438 LEA CURFINE,A0              GET ADDRESS OF CURRENT FINE
439 MOVE.B DF1ASUB(A0),D0       GET OVERALL FIELD
440 MOVE.W #DF_SEG,D1           GET DESCRIPTOR
441 JSR GETBITS                 GET SEG. NO.
442 SUBQ.W #1,D1                DECR. SEGMENT NO.
443 MOVE.W D1,D2                LOAD DATA
444 MOVE.W #DF_SEG,D1           GET DESCRIPTOR
445 JSR PUTBITS                 UPDATE FIELD
446 BSET #DF_CONT,D0            FLAG AS CONTINGUOUS
447 CMP.W #LAST,D2             CHECK IF LAST SEGMENT
448 BNE.S NTLSTN                NOT LAST SEGMENT
449 BSET #DF_RESP,D0            FLAG RESPONSE
450 MOVE.B #FALSE,PFINE         PROC A FINE <= FALSE
451 NTLSTN MOVE.B D0,DF1ASUB(A0) PUT SUBTYPE INTO CURRENT FINE
452 LEA CURDF,A1                GET ADDRESS OF CURRENT DF
453 DFPARM A0,A1                XFER PARAMS
454 BRA SENDDF                  SEND REQUEST
455 *
456 * COARSE REQUEST FOUND, USE IT
457 *
458 USECRS LEA CURDF,A2          GET ADDRESS OF CURRENT DF
459 DFPARM A1,A2                XFER PARAMS
460 ?UNLOC HBLOK
461 MOVE.B #COARSE,LASTSEG     LAST SEGMENT <= COARSE
462 BRA SENDDF                  SEND REQUEST
463 *
464 * LAST SEGMENT WAS COARSE,
465 * USE NEXT SEGMENT OF FINE
466 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

457 COARS LEA CURFINE,A0 GET ADDRESS OF CURRENT FINE
458 MOVE.B DFLASUB(A0),D0 GET OVERALL FIELD
459 MOVE.W #DF_SEG,D1 GET DESCRIPTOR
460 JSR GETBITS GET SEG. NO.
461 SUBQ.W #1,D1 DECR. SEG. NO.
462 MOVE.W D1,D2 LOAD DATA
463 MOVE.W #DF_SEG,D1 LOAD DESCRIPTOR
464 JSR PUTBITS
465 CMP.W #LAST,D2 CHECK IF LAST SEGMENT
466 BNE.S NTLSTF2 NOT LAST SEGMENT
467 BSET #DF_RESP,D0 FLAG RESPONSE
468 MOVE.B #FALSE,PFINE PROC A FINE <= FALSE
469 NTLSTF2 BCLR #DF_CONT,D0 NOT CONTIGUOUS
470 MOVE.B D0,DFLASUB(A0) UPDATE SUBTYPE IN CURRENT FINE
471 LEA CURDF,A1 GET ADDRESS OF CURRENT DF
472 DFPARM A0,A1 XFER PARAMS.
473 MOVE.B #FINE,LASTSEG LAST SEGMENT <= FINE
474 BRA SENDDF SEND REQUEST
475
476 *
477 * NOT CURRENTLY PROCESSING A FINE,
478 * TRY TO GET A FINE REQUEST
479 *
480 NOTFINE ?LOCKW HBLOCK
481 JSR GETFINE
482 TST.W D1
483 BNE USEFINE
484
485 *
486 * NO FINE REQUEST FOUND,
487 * TRY TO GET A COARSE REQUEST
488 *
489 JSR GETCRS
490 TST.W D1
491 BNE CRSFND
492
493 *
494 * NO FINE OR COARSE REQUESTS FOUND
495 * TRY TO GET A CALIB
496 *
497 LEA DFTYPTAB,A0 GET PTR TO HDR BLOCK TABLE
498 MOVE.W #TCALIB,D0 GET CALIB TYPE NO.
499 SUBQ.W #1,D0 DECR. FOR OFFSET
500 ASL.W #2,D0 MULTIPLY BY 4
501 MOVE.L 0(A0,D0.W),A0 GET PTR TO CALIB HDR BLOCK
502 JSR GETDF TRY TO GET A CALIB.
503 TST.W D0 CHECK STATUS FLAG
504 BNE USEFINE IF NOT 0, THEN FOUND ONE
505
506 *
507 * NO FINE, COARSE, OR CALIB REQUESTS
508

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

514 *      ?UNLOC HBLOK
515 *      MOVE.B #TRUE,DFIDLE
516 *      BRA SENDDF
517 *
518 *      * FOUND A COARSE, USE IT
519 *
520 *      CRSPND LEA CURDF,A2
521 *      DFPARM A1,A2
522 *      ?UNLOC HBLOK
523 *      MOVE.B #FALSE,DFIDLE
524 *      BRA SENDDF
525 *
526 *      * FOUND A FINE, USE IT
527 *
528 *      USEFINE LEA CURFINE,A2
529 *      DFPARM A1,A2
530 *      ?UNLOC HBLOK,A2
531 *      LEA CURDF,A0
532 *      DFPARM A2,A0
533 *      MOVE.B #FALSE,DFIDLE
534 *      MOVE.B DF1_TY(A0),D0
535 *      CMP.B #TINTEROP,D0
536 *      BNE.S NOTINOP
537 *
538 *      * INTEROP TYPE, SET FLAGS
539 *
540 *      MOVE.B #TRUE,INTEROP
541 *      MOVE.B #FALSE,PFINE
542 *      BRA SENDDF
543 *
544 *      * NOT INTEROP, CHECK FOR ADF BITE
545 *
546 *      NOTINOP MOVE.B #FALSE,INTEROP
547 *      CMP.B #TBITE,D0
548 *      BNE.S NOTBITE
549 *
550 *      * IT IS A BITE, CHECK FOR ADF BITE
551 *
552 *      MOVE.B DF1_ASUB(A0),D1
553 *      BTST #DF_FINE,D1
554 *      BNE.S FINBITE
555 *
556 *      * IT IS ADF BITE, SET FLAGS
557 *
558 *      MOVE.B #TRUE,PABITE
559 *      MOVE.B #TRUE,PFINE
560
  
```

NO DF REQUESTS FOUND

GET ADDRESS OF CURRENT DF
 XFER PARAMETERS

DFIDLE <= FALSE
 SEND REQUEST

GET ADDRESS OF CURRENT FINE
 XFER PARAMS.

GET ADDRESS OF CURRENT DF
 XFER PARAMS.

DFIDLE <= FALSE
 GET DF TYPE
 CHECK IF INTEROP
 NOT INTEROP

INTEROP <= TRUE
 PROC. A FINE <= FALSE
 SEND REQUEST

INTEROP <= FALSE
 CHECK IF BITE
 NOT A BITE

GET SUBTYPE
 CHECK IF ADF BITE
 NOT ADF BITE, SYSTEM BITE

PROC ADF BITE <= TRUE
 PROC A FINE <= TRUE

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

561      *      BRA      SENDDF      SEND REQUEST
562      *
563      * SYSTEM BITE
564      *
565      FINBITE  MOVE.B  #YES,SIGREQ  REQUEST SIGNAL GENERATOR
566      MOVE.B  #TRUE,PFINE  PROC. A FINE <= TRUE
567      MOVE.B  #FINE,LASTSEG  LAST SEGMENT <= FINE
568      BRA      SENDDF
569      *
570      * NOT INTEROP OR ADF BITE,
571      * CHECK IF CALIB
572      *
573      NOTBITE  CMP.B  #TCALIB,DO  CHECK IF CALIB
574      BNE.S  NOTCALB  NOT A CALIB
575      *
576      * IT IS A CALIB, SET FLAGS
577      *
578      MOVE.B  #TRUE,PCALIB  PROC A CALIB <= TRUE
579      MOVE.B  #TRUE,PFINE  PROC A FINE <= TRUE
580      MOVE.B  #YES,SIGREQ  REQUEST SIGNAL GENERATOR
581      BRA.S  SENDDF  SEND REQUEST
582      *
583      * CHECK IF AUDIO CORRELATOR REQUESTED
584      *
585      NOTCALB  MOVE.B  DF1_ASUB(A0),D1  GET SUBTYPE
586      MOVE.B  #FALSE,CFACR  ASSUME NO AUDIO CORR.
587      BTST  #DF_AUDIO,D1  CHECK IF ACORR REQ.
588      BEQ.S  NOACORR  NOT REQUESTED
589      *
590      BCLR  #DF_AUDIO,D1  CLEAR TUNE REQUEST (IN CURFINE)
591      LEA  CURFINE,A1  GET ADDR. OF CURRENT FINE
592      MOVE.B  D1,DF1_ASUB(A1)  UPDATE IN CURRENT FINE
593      *
594      * AUDIO CORR. REQUESTED, CHECK IF AVAILABLE
595      *
596      TST.B  ACORAV  CHECK IF ACORR AVAILABLE
597      BEQ.S  NOTAV  NOT AVAILABLE
598      *
599      * AUDIO CORR. REQUESTED AND AVAILABLE, SET FLAGS
600      *
601      MOVE.B  #YES,ACORREQ  AUDIO CORR REQ <= YES
602      MOVE.B  #TRUE,CFACR  CUR FINE ACORR <- TRUE
603      MOVE.B  #FALSE,ACORAV  ACORR AVAILABLE <= FALSE
604      BRA.S  NOACORR  CONTINUE
605      *
606      * NOT AVAILABLE, CLEAR TUNE ACORR RCVR FLAG
607      *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

608 MOTAV MOVE.B #NO,ACORREQ AUDIO CORR REQ <= NO
609 BCLR #DF_AUDIO,D1 DO NOT TUNE RCVR
610 MOVE.B D1,DF1_ASUB(A0) PUT BACK INTO CURRENT DF
611 LEA CURFINE,A1 GET ADDRESS OF CURRENT FINE
612 MOVE.B D1,DF1_ASUB(A1) UPDATE IN CURRENT FINE
613 MOVE.B #TRUE,FFINE PROC A FINE <= TRUE
614 MOVE.B #FINE,LASTSEG LAST SEGMENT <= FINE
615 *
616 SENDDF TST.B DFIDLE CHECK IF ANYTHING TO DO
617 BNE DONE NO DF REQUEST TO SEND
618 *
619 * CURRENT DF HAS BEEN SELECTED, SEND IT
620 *
621 JSR GETPAK GET A PACKET (A6)
622 LEA CURDF,A0 GET ADDRESS OF CURRENT DF
623 DFMS A0,A6 FORMAT DF MESSAGE
624 *
625 * ENABLE OR DISABLE DF REPORTING
626 *
627 MOVE.B DF_FLAGS(A6),D0 GET OVERALL FIELD
628 TST.W TFLAG1 CHECK IF ARF1 REPORT IS ENABLED
629 BEQ.S NOARF1 IF NOT, DISABLE
630 BSET #DF_EN1,D0 ELSE, ENABLE IT
631 BRA.S CHK2
632 NOARF1 BCLR #DF_EN1,D0 DISABLE ARF1
633 CHK2 TST.W TFLAG2 CHECK IF ARF2 REPORT IS ENABLED
634 BEQ.S NOARF2 IF NOT, DISABLE
635 BSET #DF_EN2,D0 ELSE, ENABLE ARF2
636 BRA.S DOWNREP
637 NOARF2 BCLR #DF_EN2,D0 DISABLE ARF2
638 DOWNREP MOVE.B D0,DF_FLAGS(A6) UPDATE OVERALL FIELD
639 ;
640 ; Check for calibration request. If it is a calibration DF message,
641 ; then specify the ARF id to be for ARF 1 only. This is done to
642 ; prevent calibration responses from other ARFs while calibrating
643 ; only one particular ARF.
644 ; It may have been possible to have the Perkin-Elmer specify ARF
645 ; id, but chances of that seemed slim (limitations now realized).
646 ;
647 CHPI.B #TCALIB,DF_TYP(A6) is this a cal DF request?
648 BNE.S NOTARF1 no, leave DF msg tagged for both ARFs
649 MOVE.B #S12,DF_N1B(A6) else, specify cal for ARF 1 only
650 NOTARF1 MOVE.B DF1_ASUB(A0),D0 GET SUBTYPE
651 BTST #DF_RESP,D0 CHECK IF RESPONSE
652 BEQ NOCNTL NO RESPONSE TO BE GENERATED
653 *
654 * RESPONSE WILL BE GENERATED,

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1: [ALGO. IGR. MC68000_DF]DFG3.S68_NUM

```

655 * NEED A CONTROL PACKET FOR THE RESULTS
656 *
657 MOVE.L A6,A5      GET A COPY OF DF MESSAGE ADDR.
658 JSR GETPAK        GET A CONTROL PACKET
659 MOVE.B #CTL_MID,CTL_MID(A6) PUT MESS. ID INTO CNTL
660 CLR.B CTL_FLG(A6)  CLEAR FLAGS
661 MOVE.W DF_ACC(A5),CTL_ACC(A6) PUT ACCOUNTABILITY
662 MOVE.W #TIMEOUT,CTL_TIM(A6) PUT TIME OUT AMOUNT IN
663 MOVE.B DF_ASUB(A5),D0 set coarse df timeout amount
664 ;to be 1/2 second
665 BTST #DF_FINE,D0
666 BNE.S FINE_TIME
667 MOVE.W #COARTIME,CTL_TIM(A6)
668 FINE_TIME MOVE.B DF1_POR(A0),CTL_POR(A6) MOVE PORTION NO.
669 MOVE.B DF1_DSE(A0),CTL_DSE(A6) MOVE DS ENTRY NO.
670 MOVE.L DF1_SIG(A0),CTL_SIG(A6) MOVE SIGNAL TYPE BM
671 MOVE.B DF1_ARF(A0),CTL_ARF(A6) MOVE ARF ID.
672 MOVE.B DF1_BND(A0),CTL_BND(A6) MOVE BANDWIDTH
673 MOVE.B DF1_FR1(A0),CTL_FR1(A6) 1ST BYTE OF FREQ.
674 MOVE.B DF1_FR2(A0),CTL_FR2(A6) 2ND BYTE OF FREQ.
675 MOVE.B DF1_FR3(A0),CTL_FR3(A6) 3RD BYTE OF FREQ.
676 MOVE.B DF1_FR4(A0),CTL_FR4(A6) 4TH BYTE OF FREQ.
677 MOVE.B DF1_FR5(A0),CTL_FR5(A6) 5TH BYTE OF FREQ.
678 MOVE.B DF1_ADFC(A0),CTL_ADFC(A6) AUTO DF COUNTER
679 MOVE.B DF1_ARTF(A0),CTL_ARTF(A6) ART FLAG
680 *
681 CLR.B CTL_ACR(A6)  DF^AULT TO NO ACCORR (COARSE)
682 MOVE.B DF_ASUB(A5),D0 GET SUBTYPE
683 BTST #DF_FINE,D0  CHECK IF FINE DF
684 BEQ.S NCFACR      COARSE, SKIP OVER
685 MOVE.B CFACR,CTL_ACR(A6) PUT FLAG INTO CONTROL PACKET
686 *
687 NCFACR MOVE.L A5,CTL_MES(A6) LINK DF MESSAGE INTO CTL
688 *
689 MOVE.B DF1_TYP(A0),D0 GET DF TYPE
690 CMP.B #TBITE,D0    CHECK IF BITE
691 BEQ BITRESP        DETERMINE WHICH BITE TYPE
692 *
693 * NOT A BITE REQUEST,
694 * GET RETURN JOB ADDRESS AND PRIORITY
695 * FROM TABLES
696 *
697 EXT.W D0           MAKE TYPE INTO A WORD
698 LEA RESTAB,A2      GET ADDRESS OF RETURN ADDR. TABLE
699 LEA RESPRI,A3      GET ADDR. OF RETURN JOB PRY
700 SUBQ.W #2,D0        DECR. FOR INDEX
701 MOVE.B 0(A3,D0.W),CTL_RPRI(A6) GET RETURN PRIORITY

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO.IGR.MC68000_DF]DFG3.S68 NUM

```

702 ASL.W #2,D0          CONVERT TO OFFSET
703 MOVE.L 0(A2,D0.W),CTL_RET(A6) GET RETURN ADDRESS JOR
704 *
705 * CHECK FOR INTEROP TYPE
706 *
707 MOVE.B DF1_TYP(A0),D0  GET THE TYPE
708 CMP.B #TINTEROP,D0    CHECK IF INTEROP
709 BEQ.S #FMTINP          IF YES, REFORMAT IT
710 CMP.B #TANUAL,D0      CHECK IF A MANUAL DF
711 BNE SENDCTL           NO, GO SEND IT
712 MOVE.B DPUMODE,D0     GET CURRENT GDU MODE
713 AND.B #SOC,D0        CHECK IF EITHER INTEROP MODE
714 BEQ SENDCTL          NOT INTEROP, GO SEND IT
715 MOVE.B #TRUE,INTEROP  IS INTEROP, ADJUST FLAGS
716 MOVE.B #FALSE,FINE   NOT REALLY A FINE DF
717 *
718 * TYPE IS INTEROP, REFORMAT THE MESSAGE
719 *
720 FMTINP  MOVE.W #S2112,0(A5)  SET MES. ID, PLAT., BLOCK COUNT
721 CLR.B 3(A5)          CLEAR OUT UNUSED
722 CLR.B 9(A5)          INITIALLY CLEAR PALLET AND BNDMDTH
723 MOVE.B DF_RNG(A5),D0  GET BANDWIDTH AND DET.
724 AND.B #SEQ,D0       MASK OFF BANDWIDTH CODE
725 LSR.B #5,D0         POSITION IN LS NIBBLE
726 ADDQ.B #1,D0        ADJUST CODE
727 MOVE.B D0,9(A5)      INPUT INTO MESSAGE
728 MOVE.B CTL_ACC+1(A6),2(A5) SET ACC TO LS BYTE OF ACC.
729 MOVE.W 2(A5),CTL_ACC(A6) SET CTL ACC TO MATCH MESSAGE
730 LEA 10(A5),A0       GET PTR TO START OF 2ND BLOCK
731 MOVE.W #4,D0        GET LOOP COUNT
732 CLRINP CLR.W (A0)+    CLEAR OUT PACKET
733 DBF D0,CLRINP      ENDOOP
734 BRA.S SENDCTL
735 *
736 * NORMAL BITE OR ADF BITE,
737 * FIND OUT WHICH ONE
738 *
739 BITRESP TST.B PABITE  CHECK IF ADF BITE
740 BNE.S ADFRES        ADF BITE
741 *
742 * NORMAL BITE (FINE DF)
743 *
744 MOVE.B BITPRI,CTL_RPRI(A6) GET PRTY FOR NORMAL BITE
745 MOVE.L BITRES,CTL_RET(A6) GET RETURN ADDRESS FOR NORMAL BITE
746 BRA.S SENDCTL
747 *
748 * ADF BITE

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFG3.S68_NUM

```

749 *
750 * ADRES MOVE.B ABITPRI,CTL_RPRI(A6) GET PRY FOR ADF BITE RESP.
751 *      MOVE.L ABITRES,CTL_RET(A6) GET RETURN ADDR. FOR ADF BITE
752 *
753 * SENDCTL JSR GASEND SEND CONTROL PACKET (A6)
754 *      BRA.S DONE
755 *
756 * NOCNTL JSR GASEND NO CONTROL PACKET (A6)
757 *
758 * DONE MOVEA.L (DFG3DAT1),A0 ADDR OF TIMING WORD
759 *      MOVEA.L (DFG3OUT1),A1 ADDR TO OUTPUT TIMING WORD TO
760 *      MOVE.W (DFG3MSK1),D0 MASK USED TO CLEAR TIMING BIT
761 *      EOR.W D0,(A0) CLEAR TIMING BIT
762 *      MOVE.W (A0),(A1) OUTPUT TIMING WORD
763 *      MOVEA.L (DFG3DAT2),A0 ADDR OF TIMING WORD FOR DF FLAG
764 *      MOVEA.L (DFG3OUT2),A1 ADDR TO OUTPUT TIMING WORD TO
765 *      MOVE.W (DFG3MSK2),D0 MASK USED TO CLEAR (FOR COARSE)
766 *      ; OR SET (FOR FINE) DF FLAG
767 *      CMPI.B #COARSE,(LASTSEG) IS THIS SEGMENT A COARSE?
768 *      BEQ.S FLAGCRS YES, FLAG COARSE BY CLEARING BIT
769 *      OR.W D0,(A0) ELSE, SET BIT TO INDICATE FINE DF
770 *      BRA.S OUTDFBIT OUTPUT DF FLAG BIT
771 *      NOT.W D0 COMPLEMENT MASK TO ZERO BIT
772 *      AND.W D0,(A0) CLEAR BIT TO INDICATE COARSE DF
773 *      OUTDFBIT MOVE.W (A0),(A1) OUTPUT DF FLAG BIT
774 *      MOVEM.L (SP)+,A0-A6/D0-D7 RESTORE REGISTERS
775 *      RTS
776 *      END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFG5.S68_NUM

19.10 DFG5.S68_NUM

***** Source Listing --> DFG5.S68_NUM *****

```

1  TTL      'DFG5 - SIGNAL GENERATOR CONTROL'
2  -----
3
4  *
5  *      VERSION      DATE
6  *      1          15 AUG 83
7  *      1.1        16 JAN 84 PYW - INSERT TWO TIMING BITS
8  *      1.2        3/14/84 PKE VARIOUS CORRECTIONS
9  *      1.3        4/27/84 PKE TIME OUT CHECK
10 *      1.4        8/28/84 PKE MODS FOR BITE
11 *      1.5        1/17/85 PKE CHANGE HIGH/LOW TO 150. MHZ
12 *
13 *      *INCLUDE DFG5.FMT/G
14 *
15 *      I/O DEVICES USED:
16 *      SIO-5  OUTPUT TO SIGNAL GENERATOR
17 *
18 *      ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
19 *      GADMINUC
20 *
21 *      METHOD:
22 *
23 *      This routine retrieves the incoming packet which contains
24 *      the current DF frequency to tune the Signal Generator. The
25 *      Signal Generator is turned off. After a short
26 *      delay the Generator is tuned to the current frequency contained
27 *      in the incoming packet.
28 *
29 *      *DFG5(g5freq)
30 *
31 *      *?DLINK g5freq
32 *      *?WRITE SIGGEN
33 *      *?TWAIT #2
34 *      *?WRITE SIGGEN
35 *
36 *      *JSR RLPKX
37 *      *IF (sigtime = 0)
38 *      * THEN
39 *      *   sigtime = 2
40 *      *   *SCHED dfg5t
41 *      * ELSE
42 *      *   sigtime = 2
43 *      * *ENDIF
44 *
45 *      *set incoming frequency
46 *      *turn off Signal Generator
47 *      *wait
48 *      *tune Signal Generator to
49 *      * current frequency
50 *      *release incoming packet
51 *      *is timeout routine idle
52 *
53 *      *set two second timeout
54 *      *start timeout routine
55 *
56 *      *reset timeout to two seconds

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF|DFG5.S68_NUM

```

44 -----
45 DFG5 IDNT M,X
46 OPT XDEF DFGSST
47 XREF DSSA
48 XREF RELPAK
49 XREF RCDBIN
50 XREF SIGTIME
51 XREF DFGST
52 XREF SIGREQ
53 XREF DFGSDAT1
54 XREF DFGSOUT1
55 XREF DFGMSK1
56 XREF DFGSDAT2
57 XREF DFGSOUT2
58 XREF DFGMSK2
59 XREF
60 *SINCLUDE ARTEMAS.S/G NOLIST,?DLINK,?WRITE,?WAIT,?EXIT,?SCHED
61 *SINCLUDE ARTEDATA.S/G NOLIST,RCB
62 *SINCLUDE STRC.MAC/S
63 *SINCLUDE DFF.PAC/G
64 *SINCLUDE CALSIG.PAC/G
65 *SINCLUDE DFTYPE.EQU/G
66 SECT ADMINUC
67
68 GHZ EQU 1000000000
69 HIGHLOW EQU 150000000
70 TIMEOUT EQU 2
71
72 RCVRTUNE: DC.L FMTLST
73 DC.L 0
74 OFFTUNE: DC.B $00,$01,$00,$00
75 DS.W 0
76 FMTLST: DC.B $00,$10,$00,$10
77 TUNERUP: DS.W 2
78 DS.W 0
79
80 DFGSST:
81 ?DLINK A6
82 CLR.B SIGREQ
83 MOVEA.L (DFGSDAT1),A2
84 MOVEA.L (DFGSOUT1),A3
85 MOVE.W (DFGMSK1),D3
86 OR.W D3,(A2)
87 MOVE.W (A2),(A3)
88
89 CMP.B #TBITE,DFF_TYP(A6) CHECK IF BITE OR CALIB
90 BEQ TIMEIT IF BITE, DON'T OFFTUNE

```

;SIO-5 output to Signal Generator
 ;busy/idle flag for timeout routine (DFGST)
 ;addr of timing word #1
 ;addr to output timing word #1
 ;mask to set/clear timing bit #1
 ;addr of timing word #2
 ;addr to output timing word #2
 ;mask used to set/clear timing bit #2
 HIGH/LOW BAND BREAK (150. MHZ)
 ;signal generator timeout (seconds)
 ;gain access to incoming packet (DFF)
 RESET REQUEST
 ;addr of timing word #1
 ;addr to output timing word #1 to
 ;mask used to set timing bit #1
 ;set timing bit #1
 ;output timing word #1
 #TBITE,DFF_TYP(A6) CHECK IF BITE OR CALIB
 IF BITE, DON'T OFFTUNE

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG5.S68_NUM

```

91  *
92  MOVE.L #OFFTUNE,RCVRTUNE+4 ;set data list addr for SIO parm block
93  ?WRITE DSSA,#2,RCVRTUNE,...A6 ;turn off signal generator
94  MOVEA.L (DFG5DAT2),A2 ;addr of timing word #2
95  MOVEA.L (DFG5OUT2),A3 ;addr to output timing word #2 to
96  MOVE.W (DFG5MSK2),D3 ;mask used to set timing bit #2
97  OR.W D3,(A2) ;set timing bit #2
98  MOVE.W (A2),(A3) ;output timing word #2
99  ?TWAIT #2,#1,A6
100  LEA TUNEBUF,A1 ;pointer to tuning buffer
101  LEA DFF_FR1(A6),A0 ;convert portion of incoming frequency
102  MOVE.L #10,D0 ;:- to integer to check for high/low
103  JSR RCDBIN ;:- band limit
104  CMPI.L #HIGHLOW,D1 ;high band or low band?
105  BLT.S G5LOW ;lowband
106  BSET #GEN_BAND_GEN_CTL(A1) ;set high band
107  BRA.S G5BITS ;set other bits
108  G5LOW:
109  BCLR #GEN_BAND_GEN_CTL(A1) ;set low band
110  G5BITS:
111  BSET #GEN_RF_GEN_CTL(A1) ;turn on transmitter
112  BCLR #GEN_SSB_GEN_CTL(A1) ;turn off Single Side Band
113  CMPI.L #GHZ,D1 ;frequency is GHZ
114  BLT.S G5FRQ ;:- no - set frequency
115  BSET #GEN_FR4_GEN_CTL(A1) ;set GHZ bit
116  G5FRQ:
117  MOVE.B DFF_FR1(A6),D2 ;set incoming frequency
118  LSL.L #8,D2
119  MOVE.B DFF_FR2(A6),D2 ;:-
120  LSL.L #8,D2
121  MOVE.B DFF_FR3(A6),D2 ;:-
122  LSL.L #8,D2
123  MOVE.B DFF_FR4(A6),D2 ;:-
124  LSR.L #4,D2 ;format the current frequency
125  MOVE.B D2,GEN_FR3(A1) ;:- from the incoming packet
126  LSR.L #8,D2 ;:- to the tuning buffer
127  MOVE.B D2,GEN_FR2(A1) ;:-
128  LSR.L #8,D2 ;:-
129  MOVE.B D2,GEN_FR1(A1) ;:-
130  MOVE.L #TUNEBUF,RCVRTUNE+4 ;set data list addr for SIO parm block
131  ?WRITE DSSA,#2,RCVRTUNE,...A6 ;tune to current frequency
132  MOVEA.L (DFG5DAT2),A2 ;addr of timing word #2
133  MOVEA.L (DFG5OUT2),A3 ;addr to output timing word #2 to
134  MOVE.W (DFG5MSK2),D3 ;mask used to clear timing bit #2
135  EOR.W D3,(A2) ;clear timing bit #2
136  MOVE.W (A2),(A3) ;output timing word #2
137  JSR FELPAK ;release incoming DFF message

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG5.S68_NUM

```

138      TST.W  SIGTIME      ;is timeout routine idle
139      BGT.S  G$TIMEOUT    ;no - reset timeout
140      MOVE.W #TIMEOUT,SIGTIME ;reset timeout amount (2 seconds)
141      ?SCHED DFG5T,$4,#100,$1 ;start timeout routine
142      G$TIMEOUT:
143      MOVE.W #TIMEOUT,SIGTIME ;reset timeout amount (2 seconds)
144      MOVEA.L (DFG5DAT1),A2   ;addr of timing word #1
145      MOVEA.L (DFG5OUT1),A3   ;addr to output timing word #1 to
146      MOVE.W (DFG5MSK1),D3    ;mask used to clear timing bit #1
147      EOR.W  D3,(A2)         ;clear timing bit #1
148      MOVE.W (A2),(A3)       ;output timing word #1
149      ?EXIT
150      END

```

```
***** Source Listing ==> DFC5T.S68_NUM *****
```

401

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG5T.S68_NUM

```

44 *
45 DFG5TST:
46     SUBI.W #1,SIGTIME      ;decrement timeout counter
47     BLE.S G5TOFF          ; yes - turn it off
48     ;reschedule timeout routine
49     ?SCHED DFG5T.#4,#100,#1
50     BRA.S G5TEND          ;exit routine
51 G5TOFF:
52     MOVE.L #SIGRMT,SIOBLOCK ;set format list addr for SIO parm block
53     MOVE.L #SIGOFF,SIOBLOCK+4 ;get data list addr for SIO parm block
54     ?WRITE DSSA.#2,#SIOBLOCK ;turn off signal generator
55 G5TEND:
56     ?EXIT
57     END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRG1:[ALGO.IGR.MC68000_DF]DFG6.S68_NUM

19.12 DFG6.S68_NUM

***** Source Listing ==> DFG6.S68_NUM *****

```

1  *****
2  'DFG6 - DF IMMEDIATE ROUTINE'
3  -----
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *SINCLUDE DFG6.FMT/G
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *DFG6()
31 *
32 *IF (SIGREQ = 1)
33 * THEN
34 *   call GETPAK(86pac)
35 *   format DFF packet
36 *   ?sched DFG5(86pac),#2
37 *ENDIF
38 *IF (ACORREQ = 1)
39 * THEN
40 *   call GETPAK(86pac)
41 *   format ACCR packet
42 *   ?sched DFG8(86pac),#3
43 *   ACORAV = 0

```

VERSION DATE

1 15 AUG 83

1.1 16 JAN 84 PYW - INSERT TIMING BIT

1.2 28 AUG 84 PKE - MODS FOR BITE

1.3 18 FEB 85 PYW - CLEAR THE AUDIO CORRELATOR

REQUEST FLAG HERE TO BE SURE DFG8
 WON'T BE RESCHEDULED FOR THE SAME
 REQUEST IN CASE DFG8 DOES NOT RUN.

I/O DEVICES USED:

NONE

ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:

GADMINUC

METHOD:

This routine is used to make requests to the Signal Generator and Audio Correlator control routines. The flags which indicate which of the two has been requested are checked and the appropriate control routine is scheduled. A DFF message is formatted and sent to the Signal Generator routine or an ACCR message is formatted and sent to the Audio Correlator routine.

if Signal Generator request

reserve a small packet for DFF

format Signal Generator request

send generator request

if Audio Correlator request

reserve a small packet for ACCR

format Audio Correlator request

send correlator request

indicate Correlator unavailable

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFG6.S68_NUM

```

44 *ENDIF
45 *?EXIT
46 *
47 *-----
48      TTL      'DFG6 - DF IMMEDIATE ROUTINE'
49      IDNT
50      OPT      -M
51      XDEF      DFG6ST
52      XREF      SIGREQ
53      XREF      ACORREQ
54      XREF      ACORAV
55      XREF      CURDF
56      XREF      GETPAK
57      XREF      DFG5
58      XREF      DFG8
59      XREF      DFG6DAT
60      XREF      DFG6OUT
61      XREF      DFG6MSK
62      *SINCLUDE MIDEQU.EQU/G
63      *SINCLUDE ARTEMACS.S/G  NOLIST,?SCHED,?EXIT
64      *SINCLUDE STRC.MAC/S
65      *SINCLUDE DFF.PAC/G
66      *SINCLUDE ACCR.PAC/G
67      *SINCLUDE DFL.DBS/G
68      *SINCLUDE DFTYPE.EQU/G
69      SECT      GADMINUC
70      LIST
71      *
72      DFG6ST:
73      MOVEA.L (DFG6DAT),A0      ;addr of timing word
74      MOVEA.L (DFG6OUT),A1      ;addr to output timing word to
75      MOVE.W (DFG6MSK),D0      ;mask used to set timing bit
76      OR.W D0,(A0)             ;set timing bit
77      MOVE.W (A0),(A1)         ;output timing word
78      TST.B SIGREQ              ;signal generator requested?
79      BEQ.S DFG6ACOR            ;no - try Audio Correlator
80      ;format DFF message and schedule DFG5 with it to set
81      ;up the Signal Generator timeout.
82      JSR      GETPAK           ;allocate small packet for DFF message
83      ;released in DFG5
84      MOVE.B #DFFMID,DFF_MID(A6) ;format DFF message
85      LEA      CURDF,A5         ;frequency retrieved from CURDF
86      MOVE.B DFL_TYP(A5),DFF_TYP(A6) GET DF TYPE
87      MOVE.B DFL_FR1(A5),DFF_FR1(A6) ;
88      MOVE.B DFL_FR2(A5),DFF_FR2(A6) ;
89      MOVE.B DFL_FR3(A5),DFF_FR3(A6) ;
90      MOVE.B DFL_FR4(A5),DFF_FR4(A6) ;

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG6.S68_NUM

```

91 MOVE.B DF1_FR5(A5),DF5_FR5(A6) ;
92 MOVE.B DF1_RNG(A5),DF5_RNG(A6) ;
93 ;
94 ;schedule DFG5 to control the Signal Generator
95 *
96 MOVE.L #4,D1 ASSUME SIG GEN FOR CALIB
97 CMP.B #TBITE,DF5_TYP(A6) CHECK IF SYSTEM BITE
98 BNE.S SCHED NO, IT IS CALIB
99 CLR.L D1 SET DELAY TO 0 FOR BITE
100 SCHED ?SCHED DFG5,#2,D1,#1,A6,D0/A0-A1/A5
101 DFG6ACOR:
102 TST.B ACORREQ ;Audio Correlator requested?
103 BEQ.S DFG6END ; no - exit routine
104 ;
105 ;format ACCR message and schedule DFG8 with it to set
106 ;up the Audio Correlator timeout
107 JSR GETPAK ;allocate a small packet for ACCR message
108 ;released in DFG8
109 LEA CURDF,A5 ;frequency retrieved from CURDF
110 MOVE.B #ACCRMID,ACCR_MID(A6) ;format ACCR message
111 MOVE.W DF1_ACC(A5),ACCR_ACC(A6) ;accountability
112 MOVE.B DF1_TYP(A5),ACCR_TYP(A6) ;DF type
113 MOVE.B DF1_ASUB(A5),ACCR_SUB(A6) ;DF subtype
114 ;
115 ;schedule DFG8 to control the Audio Correlator
116 ?SCHED DFG8,#3,#4,#1,A6,D0/A0-A1
117 CLR.B ACORREQ clear audio correlator request
118 DFG6END:
119 EOR.W D0,(A0) ;clear timing bit
120 MOVE.W (A0),(A1) ;output timing word
121 ?EXIT
122 END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFG8.S68_NUM

19.13 DFG8.S68_NUM

```

***** Source Listing ==> DFG8.S68_NUM *****
*****
1  TTL      'DFG8 - AUDIO CORRELATOR CONTROL'
2  *-----*
3  *
4  *
5  *   VERSION      DATE
6  *   1          15 AUG 83
7  *   1.1        16 JAN 84 PYW - INSERT TIMING BIT
8  *   1.2        10 JUL 84 PYW - change command bit to "1" to
9  *   1.3        31 AUG 84 PKE - CORRECTIONS TO FLAGS
10 *
11 *$INCLUDE DFG8.FMT/G
12 *
13 * I/O DEVICES USED:
14 *   SIO-7  OUTPUT - START AUDIO CORRELATOR
15 *   INPUT  - READ RESULTS FROM AUDIO CORRELATOR
16 *
17 * ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
18 *   GADMINUC
19 *
20 * METHOD:
21 *
22 *DFG8(G8ACCR)
23 *
24 *?DLINK g8accr
25 *call setpak(g8acc)
26 *format g8acc from g8accr
27 *call relpak(g8accr)
28 *g8acc_res = 1
29 *IF (ACORAV = 1)
30 * THEN
31 *   ACORAV = 0
32 *   ?write ACORR
33 *   ?readw ACORR
34 *   ?dlink g8res
35 *   ACORAV = 1
36 *   IF (g8res = 1)
37 *   THEN
38 *     IF (g8res = 1)
39 *     THEN
40 *       g8acc_res = 0
41 *     ENDIF
42 *   ENDIF
43 *ENDIF

;gain access to incoming message
;allocate response packet

;release incoming message
;default AC to indicate failed
;if AC is available

;indicate AC unavailable
;start Audio Correlator
;wait for correlation results
;correlator interrupt or timeout
;indicate AC available
;if READ completed successfully

;if Audio Correlation OK

;indicate AC OK

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFG8.S68_NUM

```

44 *IF (g8acc_sub = 'F')
45 * THEN
46 * IF (ACORAV = 0)
47 * THEN
48 * g8acc_res = 2
49 * ENDIF
50 * ?sched ACBITR
51 * ELSE
52 * ?sched DFPE2
53 *ENDIF
54 *
55 -----
56 DFG8 IDNT
57 OPT -M.X
58 XDEF DFG8ST
59 XREF ACORAV
60 XREF ACORREQ
61 XREF ACBITR
62 XREF DFPE2
63 XREF DS17
64 XREF DS7B
65 XREF RELPAK
66 XREF GETPAK
67 XREF DFG8DAT
68 XREF DFG8OUT
69 XREF DFG8MSK
70 *$INCLUDE ARTEMACS.S/G NOLIST,?DLINK,?WRITE,?READW,?SCHED,?EXIT
71 *$INCLUDE ARTEDATA.S/G NOLIST,RCB
72 *$INCLUDE MIDQU.EQU/G
73 *$INCLUDE STRC.MAC/S
74 *$INCLUDE ACCR.PAC/G
75 *$INCLUDE ACC.PAC/G
76 SECT GADMINUC
77 LIST
78 *
79 TMOUT EQU 50
80 BITRES EQU $OF
81 *
82 STRTAC: DC.L ONEBIT
83 DC.L ACSTROBE
84 ONEBIT: DC.W 0001
85 ACSTROBE: DC.W 0001
86 ;
87 INPUTAC: DC.L ACPMT
88 DC.L ACDATA
89 ACPMT: DC.W 0001
90 ACDATA: DS.W 1

```

;results for BITE?
 ;if BITE and AC was busy
 ;indicate AC busy
 ;schedule BITE routine
 ;schedule DF routine

;addr of timing word
 ;addr to output timing word to
 ;mask to set/clear timing bit

;Audio Correlator timeout length
 ;Bite results flag

;l=command to start audio correl.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFG8.S68_NUM

```

91 *
92 DFG8ST:
93
94 MOVEA.L (DFG8DAT).A3 ;addr of timing word
95 MOVEA.L (DFG8OUT).A4 ;addr to output timing word to
96 MOVE.W (DFG8MSK).D3 ;mask used to set timing bit
97 OR.W D3,(A3) ;set timing bit
98 MOVE.W (A3).(A4) ;output timing word
99 ?DLINK A1 ;gain access to incoming message
100 CLR.B ACORREQ RESET REQUEST FLAG
101 JSR GETPAK ;allocate packet for AC result message
102 ;move data from the incoming ACCR message to the ACC result message
103 MOVE.B #ACCMID,ACC_MID(A6)
104 MOVE.W ACOR_ACC(A1),ACC_ACC(A6)
105 MOVE.B ACOR_TYP(A1),ACC_TYP(A6)
106 MOVE.B ACOR_SUB(A1),ACC_SUB(A6)
107 EXG A1,A6 ;release the incoming ACCR message
108 JSR RELPAK ;
109 MOVE.B #1,ACC_RES(A1) ;default Audio Correlator to failed
110 ;Audio Correlator is available
111 ;perform a correlation on the current DF frequency
112 ?WRITE DS7B,#1,$STRAC...A1 ;start Audio Correlator
113 ?READW DS17,#1,$INPUTAC,$THOUT,A1 ;wait for correlation results
114 ?DLINK D1 ;correlation interrupt(1) or timeout(2)?
115 MOVE.B #1,ACORAV ;indicate Audio Correlator available
116 CMP.W #1,D1 ;READW completed successfully
117 BNE.S G8MSG ;no - timed out
118 TST.W ACDATA ;Audio Correlator OK?
119 BEQ.S G8MSG ;no - default to failed
120 CLR.B ACC_RES(A1) ;indicate audio correlation OK
121
122 G8MSG:
123 CMP.B #BITRES,ACC_SUB(A1) ;results for BITE or DF?
124 BNE.S G8DF ;DF
125 TST.B ACORAV ;if BITE, and correlator was busy
126 BNE.S G8SCHBIT ;not busy
127 MOVE.B #2,ACC_RES(A1) ;indicate Audio Correlator was busy
128
129 G8SCHBIT:
130 ?SCHED ACBITR,$5...A1 ;schedule BITE routine
131 BRA.S G8END ;
132
133 ?SCHED DFPEZ,$4...A1 ;schedule DF routine
134
135 G8DF:
136
137 G8END:
138
139 MOVEA.L (DFG8DAT).A3 ;addr of timing word
140 MOVEA.L (DFG8OUT).A4 ;addr to output timing word to
141 MOVE.W (DFG8MSK).D3 ;mask used to clear timing bit
142 OR.W D3,(A3) ;clear timing bit
143 MOVE.W (A3).(A4) ;output timing word
144 ?EXIT

```

Improved GUARDRAIL V MC68000 'DF' Files
DRCl: [ALGO.IGR.MC68000_DF]DFG8.S68_NUM

138

END

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO,IGR,MC68000 DF]DFIN.S68_NUM

19.14 DFIN.S68_NUM

```

***** Source Listing ==> DFIN.S68_NUM *****
*****
***** DFIN -- ACCEPT MESSAGES FROM ANOTHER CPU *****
*****
1  ITL
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *SINCLUDE DFIN.PMT/G
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *DFIN()
21 *
22 *save processor state for ARTE
23 *IF (BUFDON = false)
24 * THEN
25 * ?RTN
26 * ELSE
27 * BUFDON = false
28 * call pacbuf(bufd, inpac1) :move message into packet
29 * action = action address of offset in BUFDON + start of ADF
30 * TIB table
31 * clear BUFDON :reset semaphore for buffer
32 * IF action routine not on this CPU
33 * THEN
34 * ?crash
35 * ELSE
36 * ?sched action,#?,...inpac1 :schedule action routine
37 * ?RTN
38 * ENDIF
39 *
40 *ENDIF
41 *
42 *
43
OPT M

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFIN.S68_NUM

```

44  *INCLUDE ARTEMAS.S/G NOLIST,?SAVST,?SCHED,?RTN,?CRASH
45  *INCLUDE ARTEDATA.S/G NOLIST,DIB,RCB,DRCB
46
47
48  SECT  ADFUC
49  IDNT
50
51  XDEF  DFIN
52  XREF  PACBUF
53  XREF  ACPUTIB
54  XREF  RUFDF
55  XREF  RUFDFBSY
56  XREF  RUFDFDON
57  XREF  RUFDFOFF
58  XREF  DICC
59
60  PRIORITY EQU 3
61
62
63  DFIN  TST.B  BUFDFDON
64        BEQ   RETURN
65  CLR.B BUFDFDON
66  ?SAVST DICC
67  LEA   RUFDF,A5
68  JSR   PACBUF
69  MOVE.L BUFDFOFF,D6
70
71  CLR.B  BUFDFBSY
72
73  LEA   ACPUTIB,A4
74  ADDA.L D6,A4
75  CMPI.W #0,(A4)
76
77  BNE.S SCHEDULE
78  ?CRASH
79  SCHEDULE CLR.L D0
80        MOVE.B PRIORITY(A4),D0
81        ?SCHED (A4),D0,...A6
82        ?RTN  #0
83
84
85  RETURN  RTE
86
87
88  END

```

;allocate and fill packet
 start of TIB table
 ;ADF cpu input buffer
 ;done flag for ADF's buffer
 ;offset into TIB table of action
 byte offset into TIB entry table
 ; where priority of task is stored
 ; for each entry
 ; is buffer completely filled?
 ; if not, spurious interrupt - rtn
 ; clear done flag
 ; save state of processor
 ; address of ADF buffer
 ; allocate and fill packet
 ; get offset into TIB table of desired
 ; action routine
 clear inter-CPU communications
 ; semaphore
 start of TIB table
 add offset
 is the action routine on this CPU?
 ; non-zero TIB pointer = yes
 branch if yes
 if not then crash
 get register ready for priority
 priority of desired routine
 ; schedule action routine
 ; always do a continuation return
 ; in order to allow more messages
 ; to be received
 ; if a spurious interrupt occurred,
 ; return from exception because
 ; ARTE will not be invoked

Improved GUARDRAIL V MC68000 'DF' Files DRCL: [ALGO, IGR, MC68000 DF] DFOUT.S68 NUM

19.15 DFOUT.S68_NUM

```

***** Source Listing --> DFOUT.S68 NUM *****
*****
1  TTL      DFOUT - SEND MESSAGES TO ADF CPU
2
3
4
5  VERSION  DATE
6  1        1-APRIL-83
7  2        16 AUGUST-83
8  3.0      22 AUGUST 83
9
10 *SINCLUDE DFOUT.FMT/G
11
12 * I/O DEVICES USED
13 * NONE
14
15 * ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
16 * ADMINUC, ASCAROUT
17
18 * METHOD:
19
20 * The multiple entry points are used for the specific routine
21 * that is desired on the ADF CPU. At the entry point, the
22 * corresponding offset into the TIB table is calculated for the
23 * desired routine. After calculating the offset, every job will
24 * then execute the actual inter-cpu output routine.
25 * This routine executes a timeout loop until it either reserves
26 * the ADF inter-CPU buffer or times out. On timeout the new
27 * message is released and if the current CPU has the buffer reserved
28 * the ADF CPU is interrupted again to accept the last message
29 * which is still in the buffer. If the buffer is reserved before
30 * the timeout occurs, then the message packet is moved into it.
31 * Once the packet have been transferred to the buffer it is
32 * released for reuse on the current CPU and the ADF CPU is
33 * interrupted to accept the message in the buffer.
34
35 *ADFBIT: calculate offset of ADFBIT's TIB pointer in the TIB table
36 * go to DFOUT
37
38 *DFA2: calculate offset of DFA2's TIB pointer in the TIB table
39
40 *DFOUT:
41 *?dlink outpac
42 *bufftim = OUTWAIT
43 *LCP
  
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO, IGR, MC68000_DF] DFOUT.S68 NUM

```

44 * tas (bufdfbys)
45 * IF (bufdfbys already set)
46 * THEN
47 *   bufdfbys = bufdfbys - 1
48 *   ELSE
49 *     break from loop
50 *   ENDIF
51 * IF (bufdfbys = 0)
52 * THEN
53 *   call relpac(outpac)
54 *   IF (bufdfwho = CPU)
55 *   THEN
56 *     interrupt ADF CPU
57 *   ENDIF
58 *   exit routine
59 *   ENDIF
60 * ENDOOP
61 * call bufpac(outpac, bufdf)
62 * BUFDFOFF = offse
63 * BUFDFWHO = current CPU (gadmin)
64 *
65 * BUFDFDON = true
66 *
67 * interrupt ADF CPU
68 * call relpac(outpac)
69 *
70 * -----
71
72 * OPT M
73
74 * INCLUDE ARTEMACH S/G NOLIST, ?LINK, ?EXIT, ?LOCKW, ?UNLOC
75 * INCLUDE ARTEDATA S/G NOLIST, ROB
76
77 * SECT ADMINUC
78 * IDNT
79
80 * XDEF ADFBITST
81 * XDEF ?FA2ST
82
83 * XREF ADFBIT
84 * XREF ?FA2
85 * XREF ADFBITB
86 * XREF ?BUDF
87 * XREF ?BUDFBSY
88 * XREF ?BUDFDON
89
90 * XREF ?BUDFWHO

```

; attempt to get output buffer
 ; if buffer in use
 ; decrement timeout counter
 ; if timeout occurs
 ; release timed-out message
 ; if current CPU is using buffer
 ; move message to buffer
 ; pass offset into TIB table to cpu
 ; indicate cpu which has current use of
 ; buffer
 ; indicate buffer is filled (this aids
 ; the detection of spurious interrupts)
 ; release local packet

TIB pointer
 TIB pointer
 TIB table start address
 ; ADF buffers
 ; byte, busy/free
 ; byte, done filling buffer flag
 ; used to handle spurious intrts.
 ; longword, identify cpu which currently

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFOUT.S68 NUM

```

91      XREF      BUFDFOFF
92      ; holds buffer
93      ; longword, offset into TIB table of
94      ; desired task.
95      XREF      RELPAK
96      XREF      BUFFAC
97      XREF      CPUCOMFL
98      ; message moving routine
99      ; flag set if inter-CPU communications
100      XREF      ADFOLOK
101      ; timeout occurred
102      ; ADF cpu output communication
103      ; routine lock
104
105      EQU      SPFFSFA
106      EQU      125
107      EQU      5
108      ; Interrupt ADF CPU address
109      ; buffer timeout length = 0.5 msec
110      ; bit 5 indicates an error on
111      ; the ADF CPU. (CPU #5)
112
113      ADFBITST MOVE.L #ADFBIT.D6
114      SUBI.L #ACPUTIB.D6
115      BRA.S DFOUT
116
117
118      DFA2ST MOVE.L #DFA2.D6
119      SUBI.L #ACPUTIB.D6
120      ; D6 = addr of DFA2's TIB pointer
121      ; calculate offset into TIB table of
122      ; desired routine
123
124      ?DLINK A6
125      ?LOCKW ADFOLOK,A5/D6
126      ; set address of packet to be sent
127      ; lock ADF output routine
128      ; save register pointing to packet
129      ; and register containing TIB offset
130      ; initialize timeout counter
131      ; 4 usec per loop
132      ; grab buffer if not in use
133      ; buffer grabbed, send message
134      ; loop until timeout
135      BSET #ADFERR,(CPUCOMFL)
136      ; set inter CPU communications
137      ; timeout error flag for status
138      ; message
139      CMPI.L #ACPUTIB,BUFDFWHO
140      ; is current cpu using buffer
141      ; CPU's TIB table addr is used as
142      ; the identifier.
143      BNE.S ADFOEND
144      ; no - exit routine

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFOUT.S68_NUM

```

138      BRA.S      ADFODELY      ;continue testing
139      ADFODELY
140      MOVE.L      #ACPUTIB, RUFDFWHO ;current cpu is now using buffer
141      ; CPU's TIB table addr identifies it
142      LEA         RUFDF, A5      ;get address of destination buffer
143      JSR         RUFFAC         ;move message into buffer
144      MOVE.L      D6, RUFDFOFF   ;offset into TIB table for desired
145      ; task on ADF CPU
146      MOVE.B      #1, RUFDFDON   ;set done flag
147      ADFODELY    TST.B          ;interrupt ADF cpu
148      ADFODEND    ?UNLOC        ;unlock to allow next routine thru
149      JSR         HELPAK         ;release message packets
150      ?EXIT
151      END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000 DF]DFPE1.S68_NUM

19.16 DFPE1.S68_NUM

```

***** Source Listing --> DFPE1.S68_NUM *****
*****
1  TTL      'DFPE1 - RECEIVE DF REQUESTS'
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *$INCLUDE DFPE1.FMT/G
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 * PDL:
19 *
20 * ?DLINK IPACK /* GET INPUT PACKET */
21 * CALL GETPAK
22 * FORMAT AS DF TYPE /* IPACK PARAMS. TO NEW PACKET */
23 * CALL RELPAK /* RETURN INPUT PACKET */
24 * ?SCHED DFG1,#3,#0,#1,IPACK
25 * ?EXIT
26 *
27 *
28 *
29 *
30 *
31 DFPE1 IDENT
32 OPT -M
33 XDEF DFPE1E
34 XREF DFG1
35 XREF GETPAK
36 XREF RELPAK
37 *
38 *$INCLUDE STRC.MAC/S
39 *
40 *$INCLUDE DF.PAC/G
41 *
42 *$INCLUDE DF.PAC/G
43 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: {ALGO.IGR.MC68000_DF]DFPE1.S68_NUM

```

44 *SINCLUDE ARTEMAC: S/G NOLIST, ?DLINK, ?SCHED, ?EXIT
45 *
46 *SINCLUDE ARTEDATA.S/G NOLIST, RGB
47 *
48 NOLIST
49 *SINCLUDE MIDEQU EQU/G
50 LIST
51 *
52 SECT GADMINUC
53 *
54 DS.W 0 force word boundary
55 *
56 DFPE1E ?DLINK A0 ; GET INPUT PACKET
57 JSR GETPAK GET A PACKET
58 MOVE.B #DFMID, DF_MID(A6) REFORMAT PACKET
59 MOVE.W DP_ACCNT(A0), DP_ACC(A6)
60 MOVE.B DP_FR1(A0), DP_FR1(A6)
61 MOVE.B DP_FR2(A0), DP_FR2(A6)
62 MOVE.B DP_FR3(A0), DP_FR3(A6)
63 MOVE.B DP_FR4(A0), DP_FR4(A6)
64 MOVE.B DP_FR5(A0), DP_FR5(A6)
65 MOVE.B DP_GAIN(A0), DP_GAIN(A6)
66 MOVE.B DP_RNG(A0), DP_RNG(A6)
67 MOVE.B DP_FLAGS(A0), DP_FLAGS(A6)
68 MOVE.B DP_TYP(A0), DP_TYP(A6)
69 MOVE.B DP_ASUB(A0), DP_ASUB(A6)
70 ?SCHED DFG1, #3, ..., A6, A0
71 RELEASE MOVE.L A0, A6 GET READY TO CALL RELPAK
72 JSR RELPAK RETURN INPUT PACKET
73 *
74 ?EXIT
75 END

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO. IGR. MC68000 DF] DFPE2.S68_NUM

19.17 DFPE2.S68_NUM

```

***** Source Listing ==> DFPE2.S68_NUM *****
*****
***** DFPE2 - DF REPORT TO PERKIN ELMER *****
*****
1  TTL
2  *-----*
3  *
4  *
5  *   VERSION   DATE
6  *   1         8 SEPT 83
7  *   1.1       27 APR 84
8  *   1.2       14 JUNE 84
9  *   1.3       16 JUL 84   PYW - DROP DF RESPONSES WITH
10 *   1.4       9 AUG 84   LAT/LONG OF ZERO
11 *   1.5       20 AUG 84   PYW - ZERO OUT DF RESPONSE PACKET
12 *   1.6       8 SEPT 84   TO PREVENT PE FROM MISTAKING UNUSED
13 *   1.6       8 SEPT 84   FIELDS AS DATA (E.G. DF REJECTED MSG)
14 *   1.6       8 SEPT 84   MODIFICATIONS FOR INTEROP
15 *   1.6       8 SEPT 84   - VARIOUS AUDIO CORRELATION
16 *   1.6       8 SEPT 84   CORRECTIONS
17 *$INCLUDE DFPE2.FMT/G
18 *
19 *   I/O DEVICES USED:
20 *   NONE
21 *
22 *   ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
23 *   GADMINUC
24 *
25 *   METHOD:
26 *
27 *DFPE2(pe2pac)
28 *
29 *?DLINK pe2pac
30 *IF (pe2pac_MID = DFRJMID) or (pe2pac_REJ = 1) or (lat and long = 0)
31 * THEN
32 *   call getlpak(pe2resp)
33 *   format DF rejected message
34 *   ?SCHED spout,#1,#0,,pe2resp
35 *   call relpak(pe2pac)
36 * ELSE
37 *   IF (pe2pac_MID = ACCMID)
38 *   THEN
39 *   IF (DFRESULT = 0) or (DFRESULT_ACC <> pe2pac_ACC)
40 *   THEN
41 *   IF (ACRESULT <> 0)
42 *   THEN
43 *     call relpak(ACRESULT)

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFPE2.S68_NUM

```

44 *      ENDIF
45 *      ACRESULT = pe2pac
46 *      ELSE
47 *      call getlpak(pe2resp)
48 *      pe2resp_ACR = ACCPAC_RES
49 *      ENDIF
50 *      ELSE
51 *      IF (pe2pak_ACR = 1)
52 *      THEN
53 *      IF (ACRESULT = 0) or (pe2pac_ACC <> ACRESULT_ACC)
54 *      THEN
55 *      IF (DFRESULT <> 0)
56 *      THEN
57 *      call relpak(DFRESULT)
58 *      ENDIF
59 *      DFRESULT = pe2pac
60 *      ELSE
61 *      call getlpak(pe2resp)
62 *      pe2resp_ACR = ACCPAC_RES
63 *      ENDIF
64 *      ENDIF
65 *      ENDIF
66 *      ENDIF
67 *      *format DF response message
68 *      *SCHED spout,#1,#0,,pe2resp
69 *      *call relpak(pe2pac)
70 *      *IF (INTEROP = 1)
71 *      THEN
72 *      INTEROP = 0
73 *      * ?SCHED dfg3,#1,#0
74 *      *ENDIF
75 *
76 *
77 DFPE2 IDNT M
78 OPT
79 XDEF DFPE2ST
80 XREF GROUT
81 XREF GETLPK
82 XREF RELPAK
83 XREF DFG3
84 XREF INTEROP
85
86
87 XREF WRITER
88 XREF HEXSTR
89
90

```


Improved GUARDRAIL V MC68000 'DF' Files
 DR01:[ALGO.IGR.MC68000_DF]DFPE2.S68_NUM

```

91 *SINCLUDE ARTEMACS.S/G NOLIST,?DLINK,?SCHED,?EXIT
92 *SINCLUDE ARTEDATA.S/G NOLIST,RCB
93 *SINCLUDE GLOBEQU.EQU/G
94 *SINCLUDE STRC.MAC/S
95 *SINCLUDE MIDEQU.EQU/G
96 *SINCLUDE DFPR.PAC/G
97 *SINCLUDE ACC.PAC/G
98 *SINCLUDE DFRJ.PAC/G
99 *SINCLUDE DFR.PAC/G
100 *SINCLUDE CTL.PAC/G
101 *
102 *SINCLUDE DFTYPE.EQU/G
103 *
104
105 STRC
106 USES 1
107 USES 1,BLOCK
108 BITS 4,AARFID
109 BITS 4,BLKNT
110 USES 2,ACCTNUM
111 ENDST GEN
112 SECT GADMINUC
113 LIST
114 CNTLG EQU S07
115 CR EQU S0D
116 LF EQU S0A
117 MAXBL EQU 8-1
118 INTBL EQU SFFF
119 *
120 INTERRS DC.W 0
121 DC.W S0008
122 DC.W S0040
123 DC.W S0080
124 *
125 DFRESULT DC.L 0
126 ACRESULT DC.L 0
127 ERRMSK DC.W S0103
128 *
129 NAVERR1 DC.B 'ZERO LAT/LONG REPORTED FROM ARF 1',CNTLG,CR,LF,0
130 NAVERR2 DC.B 'ZERO LAT/LONG REPORTED FROM ARF 2',CNTLG,CR,LF,0
131 DS.W 0
132 force word boundary
133 *
134 *
135 DFPE2ST: ?DLINK A1 ;gain access to incoming packet
136 CMP.B #DFRJMID,(A1) ;is message a 'DF rejected'?
137

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFPE2.S68_NUM

```

138 BEQ.S PE2REJ      ; yes - format 'rejected message'
139 CMP.B #CTLMD,(A1) ; is message a CTL packet?
140 BNE PE2CORR      ; no - try correlator response
141 BTST #CTL_REJ,CTL_FLG(A1) ; was DF message timed out?
142 BEQ PE2DF        MESSAGE DID NOT TIME OUT
143 BTST #CTL_1ST,CTL_FLG(A1) ONE OR BOTH TIMED OUT, WAS
144 *                IT THE FIRST ONE ?
145 BNE PE2DF        FIRST MESSAGE CAME IN
146 BTST #CTL_2ND,CTL_FLG(A1) DID SECOND ONE TIME OUT ?
147 BNE PE2DF        SECOND MESSAGE CAME IN
148 *                ELSE, NEITHER MESSAGE CAME IN
149 PE2REJ:
150 JSR GETLPK      ; allocate packet for 'rejected' message
151 MOVEA.L A6,A0   copy packet addr to clear it out
152 MOVE.W #DFPR,D6 length of packet
153 LSR.W #2,D6     change to longword count
154 CLR.L (A0)+     clear packet...note that loop
155 DBF D6,CLEAR1   ; will go (#DFPR/4 + 1) times
156 MOVE.B #DFPRMD,(A6) ;message id (DFPR)
157 MOVE.B #1,DFPR_REJ(A6) ;indicate DF was rejected
158 MOVE.W DFRJ_ACC(A1),DFPR_ACC(A6) ;accountability
159 ?SCHED GPOUT.#1,#0,,A6,A1 ;send message to Perkin Elmer
160 CMP.B #CTLMD,0(A1) WAS INCOMING A CTL PACKET
161 BNE RELINP      NO, RELEASE IT
162 TST.B CTL_ACR(A1) WAS DF MARKED WITH ACCORR ?
163 BEQ.S RELINP    NO, RELEASE CTL PACKET
164 TST.L ACRESULT   DID THE ACRESULT COME IN ?
165 BEQ.S RELINP    NO, RELEASE CTL
166 MOVE.L ACRESULT,A6 GET AC RESULT
167 MOVE.W ACC_ACC(A6),D0 GET ACCOUNTABILITY
168 CMP.W CTL_ACC(A1),D0 DOES ACC MATCH CTL REQUEST ?
169 BNE.S RELINP    NO, RELEASE CTL
170 JSR RELPAK      YES, RELEASE ACC MESSAGE
171 CLR.L ACRESULT   RESET PACKET PTR
172 RELINP MOVE.L A1,A6 RELEASE CTL PACKET
173 JSR RELPAK
174 BRA PEZINTP    ;exit routine
175 *
176 * MESSAGE WAS AN AUDIO CORRELATOR RESULTS MESSAGE
177 *
178 * C(A1) <- ACC PACKET
179 *
180 PE2CORR:
181 CMP.B #ACCMID,(A1) ;message Audio Correlator results?
182 BEQ.S A'ORRES     YES, PROCESS ACC MESSAGE
183 MOVE.L A1,A6      PACKET WAS NEITHER ACC,CTL OR DFRJ
184 JSR RELPAK        GET RID OF IT

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000 DF]DFPE2.S68 NUM

```

185      BRA      PE2END
186      ACORRES  TST L DFRESULT
187      BEQ L    PE2ACRES
188      MOVE.W   ACC_ACC(A1),D1
189      MOVEA.L  DFRESULT,A2
190      CMP.W    CTL_ACC(A2),D1
191      BEQ S    PE2ACR1
192      PE2ACRES:
193      TST.L    ACRESULT
194      BEQ S    PE2ACSAV
195      MOVEA.L  ACRESULT,A6
196      JSR      HELPAK
197      PE2ACSAV:
198      MOVE.L   A1,ACRESULT
199      BRA      PE2END
200      PE2ACR1:
201      JSR      GETLPK
202      MOVEA.L  A6,A0
203      MOVE.W   #DFPR,D6
204      LSR.W    #2,D6
205      CLR.L    (A0)+
206      DBF      D6,CLEAR2
207      MOVE.B   ACC_RES(A1),DFPR_ACR(A6) ;insert AC response in message
208      EXG      A1,A6
209      JSR      HELPAK
210      EXG      A1,A6
211      MOVEA.L  DFRESULT,A1
212      CLR.L    DFRESULT
213      BRA.S    PE2FORM
214      *
215      * MESSAGE WAS A DF RESULTS MESSAGE
216      *
217      * C(A1) <- CTL PACKET
218      *
219      PE2DF:
220      CMP.B    #1,CTL_ACR(A1) ;DF marked for audio correlation
221      BNE.S    PE2FRM
222      TST.L    ACRESULT
223      BEQ S    PE2DFRES
224      MOVEA.L  CTL_ACC(A1),D1
225      MOVEA.L  ACRESULT,A2
226      CMP.W    ACC_ACC(A2),D1
227      BEQ S    PE2ACR2
228      PE2DFRES:
229      TST.L    DFRESULT
230      BNE.S    PE2DFSAV
231      MOVEA.L  A1,DFRESULT

```

```

EXIT
;has DF response been received?
;no - save correlator response
;do correlator and DF
;- responses match
;-
;yes - insert correlator response

;audio correlation response exists?
;no - save one
;release the old correlation
;- response

;save Audio correlator response
;exit routine

;allocate DF response packet
copy packet addr to clear it out
length of packet
change to longword count
clear packet...note that loop
;will go (#DFPR/4 + 1) times
;release Audio Correlator response
;-

;set matching DF message
RESET PACKET PTR
;format message

```

```

;DF marked for audio correlation
;no - format DF response
;has correlation result been received
;no - save DF response
;do DF and Audio Correlator
;- responses match
;-
;yes - insert correlator response

;DF response exists?
RESPONSE EXISTED
INPUT NEW RESULT

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFPE2.S68_NUM

```

232      BRA      PE2END      EXIT (WAIT FOR ACC)
233      PE2DFSAB:
234      MOVE.L   DFRESULT,A6      GET OLD DF RESULT
235      MOVE.L   A1,DFRESULT      INPUT NEW DF RESULT
236      MOVE.L   A6,A1
237      BRA.S    PE2FRM          GO FORMAT THE DF RESPONSE FOR PE
238      PE2ACR2:
239      JSR      GETLPK          ;allocate DF response packet
240      MOVEA.L  A6,A0          copy packet addr to clear it out
241      MOVE.W   #DFPR,D6      length of packet
242      LSR.W    #2,D6          change to longword count
243      CLR.L    (A0)+          clear packet...note that loop
244      DBF      D6,CLEAR3      ; will go (#DFPR/4 + 1) times
245      MOVE.B   ACC_RES(A2),DFPR_ACR(A6) ;insert AC response in message
246      EXG      A2,A6          ;release saved Audio Correlator response
247      JSR      RELPAK          ;
248      CLR.L    ACRESULT      ;
249      EXG      A2,A6          ;
250      *
251      BRA.S    PE2FORM
252      *
253      * C(A1) <- CTL PACKET WITH DF RESPONSE(S)
254      *
255      PE2FRM      JSR      GETLPK          copy packet addr to clear it out
256      MOVEA.L  A6,A0          length of packet
257      MOVE.W   #DFPR,D6      change to longword count
258      LSR.W    #2,D6          clear packet...note that loop
259      CLR.L    (A0)+          ; will go (#DFPR/4 + 1) times
260      DBF      D6,CLEAR4
261      *
262      * C(A1) <- CTL PACKET WITH DF RESPONSE(S)
263      *
264      * C(A6) <- DFPR PACKET FOR PE
265      *
266      PE2FORM:
267      ;format DF response to send to the Perkin Elmer
268      MOVE.B   #DFPRMID,DFPR_MID(A6) ;insert message id
269      CLR.B    DFPR_REJ(A6)          ;indicate not rejected
270      MOVE.W   CTL_ACC(A1),DFPR_ACC(A6) MOVE ACCOUNTABILITY
271      MOVE.B   CTL_FR1(A1),DFPR_FR1(A6) ;move frequency to report
272      MOVE.B   CTL_FR2(A1),DFPR_FR2(A6) ;
273      MOVE.B   CTL_FR3(A1),DFPR_FR3(A6) ;
274      MOVE.B   CTL_FR4(A1),DFPR_FR4(A6) ;
275      MOVE.B   CTL_FR5(A1),DFPR_FR5(A6) ;
276      CLR.B    DFPR_AR1(A6)          ;zero id for ARF 1 responses
277      BTST     #CTL_IST,CTL_FLG(A1)  ;is there an ARF 1 response
278      BEQ      PE2RES2              ; no - check ARF 2

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFPE2.S68.NUM

```

279 MOVE.L CTL_RES1(A1),A2 ;retrieve ARF 1 DF response
280 CMP.B #DFEMID,0(A2) CHECK IF INTEROP OR NOT
281 BNE INT1 NOT A DFR SO INTEROP
282 MOVE.W DFR_ERR(A2),D0 GET ERROR CODE FROM ARF 1
283 AND.W ERRMSK,D0 CHECK IF NO RETURN ERRORS
284 BNE PE2RES2 ERRORS PRESENT, DON'T USE ARF 1
285
286
287
288 TST.L DFR_LAT(A2) is latitude zero?
289 BNE.S LL10K no, nav data from ARF 1 ok
290 TST.L DFR_LON(A2) is longitude zero?
291 BNE.S LL10K no, have good nav data
292 MOVEA.L A6,A4 else, send bad lat/long msg to
293 LEA NAVERR1,A6 ;terminal after saving DFR addr
294 JSR WRITER
295 MOVEA.L A4,A6 restore pointer to DFR msg
296 BRA PE2RES2 don't use ARF 1's results
297 TST.L DFR_BASE(A2) if the 1st four baselines are
298 ;zero then the error flag is
299 ;not being recognized by the
300 ;PE so set DF rejected flag
301 BNE.S BSL10K
302 TST.L DFR_BASE+4(A2)
303 BEQ PE2RES2
304 BSL10K
305
306 MOVE.B #ARF1,DFPR_AR1(A6) ;set ARF id to ARF 1
307 MOVE.W DFR_ERR(A2),DFPR_ER1(A6) ;move DF response data to the
308 MOVE.B DFR_TYP(A2),DFPR_TY1(A6) ; DF report
309 MOVE.B DFR_SSM(A2),DFPR_SS1(A6) ;
310 MOVE.L DFR_LAT(A2),DFPR_LA1(A6) ;
311 MOVE.L DFR_LON(A2),DFPR_LO1(A6) ;
312 MOVE.W DFR_LOB(A2),DFPR_LB1(A6) ;
313 MOVE.W DFR_QUAL(A2),DFPR_QU1(A6) ;
314 MOVE.W DFR_HEA(A2),DFPR_HE1(A6) ;
315 MOVE.W DFR_ROLL(A2),DFPR_RO1(A6) ;
316 MOVE.L DFR_BASE(A2),DFPR_BA1(A6) ;
317 MOVE.L DFR_BASE+4(A2),DFPR_BA1+4(A6) ;
318 MOVE.L DFR_BASE+8(A2),DFPR_BA1+8(A6) ;
319 MOVE.L DFR_BASE+12(A2),DFPR_BA1+12(A6) ;
320 BRA PE2RES2
321
322 *
323 INT1
324 MOVE.B #ARF1,DFPR_AR1(A6)
325 MOVE.B #TINTEROP,DFPR_TY1(A6) SET TYPE TO INTEROP
326 CLR.B DFR_SS1(A6) CLEAR OUT SIGNAL STRENGTH
327 MOVE.L 14(A2),D0 GET LATITUDE

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000 DF]DFPE2.S68_NUM

```

326 ASL.L #8,D0          SHIFT TO 32 BIT BAM
327 MOVE.L D0,DFPR_LA1(A6) PUT INTO RESPONSE
328 MOVE.L 18(A2),D0      GET LONGITUDE
329 CLR.B D0              SET LS BYTE TO 0
330 MOVE.L D0,DFPR_LO1(A6) PUT INTO RESPONSE
331 MOVE.W 22(A2),DFPR_LB1(A6) GET AOA
332 MOVE.W 24(A2),DFPR_QU1(A6) GET QUALITY
333 MOVE.W 26(A2),DFPR_HF1(A6) GET HEADING
334 MOVE.W 28(A2),DFPR_RO1(A6) GET ROLL ANGLE
335 MOVE.B 4(A2),D0       GET OVERALL FIELD
336 AND.W #S03,D0        MASK OFF ERROR BITS
337 ASL.W #1,D0           ADJUST FOR OFFSET
338 LEA INTERRS,A0        GET PTR TO ERROR TABLE
339 MOVE.W 0(A0,D0.W),DFPR_ER1(A6) PUT ERROR INTO RESP.
340 MOVE.W #MAXBL,D0      GET LOOP COUNT
341 LEA DFPR_BA1(A6),A0   GET PTR TO BASELINES
342 MOVE.W #INTBL,(A0)+  FILL UP BLS
343 DBF D0,INTLP1        ENDLOOP
344 *
345 PEZRES2:
346 CLR.B DFPR_AR2(A6)    ;zero id for ARF 2 response
347 BTST #CTL_2ND_CTL_FLG(A1) ;is there an ARF 2 response
348 BEQ DFP2RE0           ; no - send packet
349 MOVE.L CTL_RES2(A1),A3 ;set pointer to ARF 2 DF response
350 CMP.B #DFRMID,0(A3)   CHECK IF INTEROP OR NOT
351 BNE INT2              NOT DFR SO INTEROP
352 MOVE.W DFR_ERR(A3),D0 GET ERROR CODE
353 AND.W FRMSK,D0        CHECK IF NO RETURN ERRORS
354 BNE DFP2RE0           ERRORS PRESENT, DON'T USE ARF 2
355 *****
356
357 TST.L DFR_LAT(A3)     is latitude zero?
358 BNE.S LL20K          no, nav data from ARF 2 ok
359 TST.L DFR_LON(A3)    is longitude zero?
360 BNE.S LL20K          no, have good nav data
361 MOVEA.L A6,A4        else, zero lat/long
362 LEA NAVERR2,A6       after saving DFPE pointer, send msg
363 JSR WRITER           ; to terminal for ARF 2
364 MOVEA.L A4,A6         restore DFPE msg pointer
365 BRA DFP2RE0          don't use ARF 2's response
366 TST.L DFR_BASE(A3)   if the 1st four baselines are
367 BNE.S RSL20K         ;zero then the error flag is
368 TST.L DFR_BASE+(A3)  ;not being recognized by the
369 BEQ DFP2RE0          ;PE so set DF rejected flag
370
371
372

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO.IGR.MC68000_DF]DFPE2.S68 NUM

```

373 BSL20K
374
375 -----
376 MOVE.B #ARF2,DFPR_AR2(A6) ;set ARF id to ARF 2
377 MOVE.W DFR_ERR(A3),DFPR_ER2(A6) ;move DF response data to the
378 MOVE.B DFR_TYP(A3),DFPR_TY2(A6) ; DF report
379
380 MOVE.B DFR_SSM(A3),DFPR_SS2(A6) ;
381 MOVE.L DFR_LAT(A3),DFPR_LA2(A6) ;
382 MOVE.L DFR_LOW(A3),DFPR_LO2(A6) ;
383 MOVE.W DFR_LOB(A3),DFPR_LB2(A6) ;
384 MOVE.W DFR_QUAL(A3),DFPR_QU2(A6) ;
385 MOVE.W DFR_HEA(A3),DFPR_HE2(A6) ;
386 MOVE.W DFR_ROLL(A3),DFPR_RO2(A6) ;
387 MOVE.L DFR_BASE(A3),DFPR_BA2(A6) ;
388 MOVE.L DFR_BASE+4(A3),DFPR_BA2+4(A6) ;
389 MOVE.L DFR_BASE+8(A3),DFPR_BA2+8(A6) ;
390 MOVE.L DFR_BASE+12(A3),DFPR_BA2+12(A6) ;
391 BRA DFP2RE0
392
393 *
394 INT2
395 MOVE.B #ARF2,DFPR_AR2(A6)
396 MOVE.B #TINTEROF,DFPR_TY2(A6) SET TYPE TO INTEROP
397 CLR.B DFR_SS2(A6) CLEAR OUT SIGNAL STRENGTH
398 MOVE.L 14(A3),D0 GET LATITUDE
399 ASL.L #8,D0 SHIFT TO 32 BIT BAM
400 MOVE.L D0,DFPR_LA2(A6) PUT INTO RESPONSE
401 MOVE.L 18(A3),D0 GET LONGITUDE
402 CLR.B D0 SET LS BYTE TO 0
403 MOVE.L D0,DFPR_LO2(A6) PUT INTO RESPONSE
404 MOVE.W 22(A3),DFPR_LB2(A6) GET AOA
405 MOVE.W 24(A3),DFPR_QU2(A6) GET QUALITY
406 MOVE.W 26(A3),DFPR_HE2(A6) GET HEADING
407 MOVE.W 28(A3),DFPR_RO2(A6) GET ROLL ANGLE
408 MOVE.B 4(A3),D0 GET OVERALL FIELD
409 AND.W #903,D0 MASK OFF ERROR BITS
410 ASL.W #1,D0 ADJUST FOR OFFSET
411 LEA INTERRS,A0 GET PTR TO ERROR TABLE
412 MOVE.W 0(A0,D0.W),DFPR_ER2(A6) PUT ERROR INTO RESP.
413 MOVE.W #MAXBL,D0 GET LOOP COUNT
414 LEA DFR_BA2(A6),A0 GET PTR TO BASELINES
415 MOVE.W #INTBL,(A0)+ FILL UP BLS
416 DBF D0,INTLP2 ENDLOOP
417
418 *
419 DFP2RE0:
420 TST.B DFR_AR1(A6) CHECK IF THERE WAS AN ARF 1
421 BNE.S BENDIT YES, SEND MESSAGE
422 TST.B DFR_AR2(A6) CHECK IF THERE WAS AN ARF2

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: [ALGO.IGR.MC68000_DF]DFPE2.S68_NUM

```

420      BNE.S    SENDIT      YES. SEND MESSAGE
421      MOVE.B   #1,DFPR.REJ(A6) NEITHER ARF1 OR ARF2
422      ;schedule GROUT with the formatted DF report to send it
423      ;to the Perkin Elmer
424      SENDIT    ?SCHED GROUT,#1,...A6,A1
425      MOVEA.L   A1,A6      ;release incoming packet
426      JSR      HELPAK      ;
427      ;if an interop DF has been received reset the interop flag
428      ;so the DF scheduler will run
429      TST.B     INTEROP    ;interop DF received?
430      BEQ.S     PE2INTP    ;exit routine
431      CLR.B     INTEROP    ;reset interop DF flag
432      JSR      DFG3
433      PE2END:
434      ?EXIT
435      END

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRCl:[ALGO.IGR.MC68000 DF]DFPON.S68_NUM

DFPON.S68_NUM

19 18

***** Source Listing ==> DFPON.S68_NUM *****

```

1  TTL      'DFPON'
2  *
3  *-----*
4  *
5  *      VERSION      DATE
6  *      1.0          7/8/83
7  *
8  *SINCLUDE DFPON.FMT/G
9  *      DFPON
10 *      POWER ON INITIALIZATION FOR DF
11 *      INITIALIZE FDFC AND SIO (-3)
12 *
13 *      CALLING SEQUENCE:
14 *
15 *      JSR      DFPON
16 *
17 *
18 *      I/O DEVICES USED:
19 *
20 *      ALL CODE SECTIONS WHERE THIS MODULE IS FOUND:
21 *      ADFUC
22 *
23 *      METHOD:
24 *      PDL:
25 *
26 *      CALL ?DVRCB(<DRCB PARAMETERS>) /* SET UP DRCB FOR CONTINUATOR */
27 *      INITIALIZE DF CONTROLLER COUNTERS
28 *      CALL ACUINIT /* SET UP PARALLEL OUTPUT FOR ACUS */
29 *      CALL INITDF /* INITIALIZE FLAGS, RESET FDFC */
30 *
31 *
32 *
33 *-----*
34 *
35 *      IDNT      DFPON
36 *      XDEF      DFPON
37 *      XREF      ?DVRCB
38 *      XREF      INITDF
39 *      XREF      DFDIB
40 *      XREF      ACUINIT
41 *
42 *      FDFC REGISTERS
43 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000 DF]DFPON.S68 NUM

44	DFIPORT	DS.W	1	ORG 3FFFF800	DF INSTRUCTION REGISTER
45	DFAPORT	DS.W	1		ACCUMULATION REGISTER
46	IFAIC	DS.W	1		IF ANALOG IFACE CONTROL
47	DFRPORT	DS.W	1		DF RESET REGISTER
48		DS.W	1		UNUSED
49		DS.W	1		UNUSED
50	DFCONT	DS.W	1		DF INTERRUPT CONTROL REGISTER
51		DS.W	1		UNUSED
52	DFACCUM	DS.W	1		ACCUMULATION FIFO OUTPUT
53	DFSTA	DS.W	1		FDPC STATUS
54		DS.W	1		IF ANALOG IFACE DATA
55	IFAID	DS.W	1		
56					
57	DFDIS	EQU	30040		INTERRUPT DISABLE MASK
58					
59					
60					
61					
62					
63	DATAREG	DS.W	1	ORG 3FFFF4C0	SIO - 3
64	OUTCSREG	DS.W	1		R/W DATA REGISTER
65	ACCSREG	DS.W	1		OUTPUT CHANNEL CONTROL/STATUS REG
66	INCSREG	DS.W	1		ACCESS CONTROL REG
67	DATAACREG	DS.W	1		INPUT CHANNEL CONTROL REG
68		DS.W	1		DATA CONTROL REG
69	RESETREG	DS.W	1		UNUSED
70	INTCREG	DS.W	1		SIO CARD RESET
71					INTERRUPT CONTROL REG
72					
73					
74	DFPON	MOVEM.L	D0-D7/A0-A6,-(SP)		SAVE ALL REGISTERS
75					
76					
77					
78					
79					
80					
81					
82					
83					
84					
85					
86					
87					
88					
89					
90					

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFPON.S68_NUM

```

91 *
92 * the following sequence initializes the DF controller counters
93 *
94     MOVE.W #DFDIS,DFCONT
95     MOVE.W #SA000,OUTCSREG
96     MOVE.W #SEFFF,OUTCSREG      *output channel control for SIO
97     MOVE.W #S2000,OUTCSREG
98     MOVE.W #S6FFF,OUTCSREG      *output channel control for SIO
99     MOVE.W #S114D,DFIPORT      *acu=1, rfp=1, rcv=1
100    WAITIN: BTST    #0,DFSTA+1  *test IRP of FDF status
101             BEQ.S  WAITIN
102 *
103     JSR    ACUINIT             INITIALIZE THE PIO FOR ACUS
104 *
105     JSR    INITDF             *initialize DF interface
106 *
107     MOVEM.L (SP)+,D0-D7/A0-A6 RESTORE REGISTERS
108     RTS
109     END

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFRPT.S68_NUM

```

44 * THEN
45 *   ACCNT <- UDR1 ACC      /* GET ACCOUNTABILITY OF UDR1 */
46 *   IF UDR2 ARF1 <> NULL
47 *   THEN
48 *     IF ACCNT = UDR2_ACC /* CHECK IF SAME DF */
49 *     THEN
50 *       IF UDR3 ARF1 <> NULL
51 *       THEN
52 *         IF ACCNT = UDR3_ACC /* CHECK IF SAME DF */
53 *         THEN
54 *           OFFSET <- ARF1
55 *           GO TO GOREPORT
56 *         ENDIF
57 *       ENDIF
58 *     ENDIF
59 *   ENDIF
60 *   IF UDR1 ARF2 <> NULL
61 *   THEN
62 *     ACCNT <- UDR1 ACC
63 *     IF UDR2 ARF2 <> NULL
64 *     THEN
65 *       IF ACCNT = UDR2_ACC /* CHECK IF SAME DF */
66 *       THEN
67 *         IF UDR3 ARF2 <> NULL
68 *         THEN
69 *           IF ACCNT = UDR3_ACC
70 *           THEN
71 *             OFFSET <- ARF2
72 *             GO TO GOREPORT
73 *           ENDIF
74 *         ENDIF
75 *       ENDIF
76 *     ENDIF
77 *   ENDIF
78 *   ELSE /* REPORTING NOT ENABLED */
79 *     GO TO DONE
80 *   ENDIF
81 * ENDIF
82 *
83 * GOREPORT:
84 * REPORT IN PROGRESS <- TRUE /* SET FLAG */
85 * PTR <- ADDR. OF UDR1 ARF1
86 * UDR1BASE <- C(FTR + OFFSET)
87 * FORMAT ERROR CODE,TYPE,SIGNAL STRENGTH,LAT.,LONG.,QUALITY
88 *   HEADING, AND ROLL ANGLE
89 * CALL WRITEH(OUTPUT STRIN:)
90 * CALL GETPRMPT

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFRPT.S68 NUM

```

91 * FORMAT BASELINE VALUES
92 * CALL WRITEH(OUTPUT STRING)
93 * CALL GETPRMPT
94 * CALL RELPAK(UDR1BASE)
95 * C(PTR+OFFSET) <- NULL
96 *
97 * PTR <- ADDR. OF UDR2 ARF1
98 * UDR2BASE <- C(PTR + OFFSET)
99 * FORMAT 5 ERROR2 ARRAY ELEMENTS
100 * CALL WRITEH(OUTPUT STRING)
101 * CALL GETPRMPT
102 * FORMAT NEXT 4 ERROR2 ARRAY ELEMENTS
103 * CALL WRITEH(OUTPUT STRING)
104 * CALL GETPRMPT
105 * CALL RELPAK(UDR2BASE)
106 * C(PTR + OFFSET) <- NULL
107 *
108 * PTR <- ADDR. OF UDR3 ARF1
109 * UDR3BASE <- C(PTR + OFFSET)
110 * FORMAT 4 ERROR2 ARRAY ELEMENTS
111 * CALL WRITEH(OUTPUT STRING)
112 * CALL GETPRMPT
113 * FORMAT SEC. MIN. VALUE, SEC. MIN. INDEX, 1ST MIN. VALUE, 1ST
114 * MIN. INDEX, RAD BASELINE NUMBER, FREQUENCY.
115 * CALL WRITEH(OUTPUT STRING)
116 * CALL GETPRMPT
117 * CALL RELPAK(UDR3BASE)
118 * C(PTR + OFFSET) <- NULL
119 * REPORT IN PROGRESS <- FALSE /* RESET FLAG */
120 *
121 * DONE: ?EXIT
122 *
123 *
124 * -----
125 *
126 * OPT M
127 * DFRPT IDNT
128 *
129 * XDEF DFRPTCT
130 * XDEF QUIT
131 * XDEF PEP
132 * XDEF TFLAG1,TFLAG2
133 * XREF RELPAK
134 * XREF GETBITS
135 * XREF HEXSTR
136 * XREF WRITEH
137 * XREF GETPRMPT

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFRPT.S68_NUM

138	*	*SINCLUDE ARTEMACS.S/G NOLIST,?DLINK,?SCHED,?EXIT	
139	*		
140	*	*SINCLUDE ARTEDATA.S/G NOLIST,RCB	
141	*		
142	*	*SINCLUDE STRC.PAC/S	
143	*		
144	*	*SINCLUDE UDR1.PAC/G	
145	*		
146	*	*SINCLUDE UDR2.PAC/G	
147	*		
148	*	*SINCLUDE UDR3.PAC/G	
149	*		
150	*		
151		NOLIST	
152	*	*SINCLUDE MIDEQU.EQU/G	
153		LIST	
154		SECT GADMINUC	
155	*		
156	ARF1	EQU	1
157	ONEBYTE	EQU	2
158	TWOBYTES	EQU	4
159	FOURBYTE	EQU	8
160	TERM	EQU	0
161	CR	EQU	\$OD
162	BLNK	EQU	\$20
163	NULL	EQU	-1
164	NBL	EQU	8
165	*		
166	ARF1ID	DC.L	1
167		DC.L	2
168	*		
169	OBUF	DS.B	R0
170		DC.B	0
171	*		
172	QUIT	DC.B	0
173	*		
174	UDR1ME1	DC.L	NULL
175	UDR1ME2	DC.L	NULL
176	UDR2ME1	DC.L	NULL
177	UDR2ME2	DC.L	NULL
178	UDR3ME1	DC.L	NULL
179	UDR3ME2	DC.L	NULL
180	*		
181	TFLAG1	DC.W	0
182	TFLAG2	DC.W	0
183	REPIP	DC.W	0
184	*		

		NO. OF DIGITS FOR ONE BYTE
		NO. OF DIGITS FOR TWO BYTES
		NO. OF DIGITS FOR FOUR BYTES
		MAX NUMBER OF BASELINES
		ARF 1 ID.
		ARF 2 ID.
		EXTRA TERMINATE
		FLAG FOR WRITEH
		CURRENT UDR1 MES. ARF 1
		CURRENT UDR1 MES. ARF 2
		CURRENT UDR2 MES. ARF 1
		CURRENT UDR2 MES. ARF 2
		CURRENT UDR3 MES. ARF 1
		CURRENT UDR3 MES. ARF 2
		REPORT FLAG FOR ARF 1
		REPORT FLAG FOR ARF 2
		REPORT IN PROGRESS FLAG

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000 DF]DFRPT.S68 NUM

185	DFRPTST	?DLINK	A5	GET INPUT PACKET
186	LEA	UDR1ME1,A0		ASSUME UDR1 ARF 1
187	CHP.B	#UDR1MID,0(A5)		CHECK IF UDR1
188	BEQ.S	UPDMES		IF SO, GO UPDATE
189	LEA	UDR2ME1,A0		ASSUME JDR2 ARF 1
190	CHP.B	#UDR2MID,0(A5)		CHECK IF UDR2
191	BEQ.S	UPDMES		IF SO, GO UPDATE
192	LEA	UDR3ME1,A0		ASSUME UDR3 ARF 1
193	*			
194	UPDMES	MOVE.B	UDR1_NIB(A5),D0	GET OVERALL FIELD
195	MOVE.W	#UDR1 ARF.D1		GET DESCRIPTOR
196	JSR	GETBITS		GET ARF ID.
197	*			
198	CHP.W	#ARF1,D1		CHECK IF ARF 1
199	BEQ.S	USEIT		IT IS ARF 1
200	ADD.L	#4,A0		ADJUST FOR ARF 2
201	*			
202	USEIT	TST.L	0(A0)	CHECK IF FIRST PACKET
203	BPL.S	NOTFIRST		NOT, FIRST TIME
204	MOVE.L	A5,0(A0)		UPDATE CURRENT PACKET
205	BRA.S	CHKREP		
206	*			
207	NOTFIRST	MOVE.L	0(A0),A6	GET OLD PACKET PTR
208	JSR	RELPAK		RELEASE OLD PACKET
209	MOVE.L	A5,0(A0)		UPDATE WITH NEW PTR
210	*			
211	CHKREP	TST.W	TFLAG1	SEE IF REPORT IS ENABLED
212	BEQ	TRYARF2		NOT FOR ARF1, CHECK ARF 2
213	*			
214	TST.L	UDR1ME1		CHECK IF A UDR1 FOR ARF 1
215	BMI	TRYARF2		NO UDR1
216	MOVE.L	UDR1ME1,A0		GET PACKET PTR FOR UDR1
217	MOVE.W	UDR1_ACC(A0),D0		GET ACCOUNTABILITY
218	TST.L	UDR2ME1		CHECK IF A UDR2 FOR ARF 1
219	BMI	TRYARF2		NO UDR2
220	MOVE.L	UDR2ME1,A0		GET PACKET PTR FOR UDR2
221	CHP.W	UDR2_ACC(A0),D0		CHECK IF SAME DF
222	BNE	TRYARF2		NOT SAME DF
223	TST.L	UDR3ME1		CHECK IF A UDR3 FOR ARF 1
224	BMI	TRYARF2		NO UDR3
225	MOVE.L	UDR3ME1,A0		GET PACKET PTR FOR UDR3
226	CHP.W	UDR3_ACC(A0),D0		CHECK IF SAME DF
227	BNE	TRYARF2		NOT SAME DF
228	*			
229	*	HAVE ALL THREE PACKETS FOR ARF1		
230	*	GET READY TO REPORT		
231	*			

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL:[ALGO.IGR.MC68000_DF]DFRPT.S68 NUM

232	CLR.W	D6	SET OFFSET FOR ARF 1
233	BRA	GOREPORT	
234	*		
235	*	CHECK IF CAN REPORT FOR ARF 2	
236	*		
237	TRYARF2	TFLAG2	CHECK IF ENABLED FOR ARF 2
238	BEQ	DONE	NEITHER ARF1 OR ARF 2
239	TST.L	UDR1ME2	CHECK IF A UDR1 FOR ARF 2
240	BMI	DONE	NO UDR1
241	MOVE.L	UDR1ME2,A0	GET PACKET PTR FOR UDR1
242	MOVE.W	UDR1_ACC(A0),D0	GET ACCOUNTABILITY
243	TST.L	UDR2ME2	CHECK IF A UDR2 FOR ARF 2
244	BMI	DONE	NO UDR2
245	MOVE.L	UDR2ME2,A0	GET PACKET PTR FOR UDR2
246	CMF.W	UDR2_ACC(A0),D0	CHECK IF SAME DF
247	BNE	DONE	NOT SAME DF
248	TST.L	UDR3ME2	CHECK IF A UDR3 FOR ARF 2
249	BMI	DONE	NO UDR3
250	MOVE.L	UDR3ME2,A0	GET PACKET PTR FOR UDR3
251	CMF.W	UDR3_ACC(A0),D0	CHECK IF SAME DF
252	BNE	DONE	NOT SAME DF
253	*		
254	*	HAVE ALL THREE PACKETS FOR ARF 2	
255	*	GET READY TO REPORT	
256	*		
257	MOVE.W	#4,D6	SET OFFSET FOR ARF 2
258	*		
259	GOREPORT	MOVE.W #1,REIP	INDICATE REPORT IN PROGRESS
260	*		
261	LEA	UDR1ME1,A3	GET ADDR. OF UDR1 FOR ARF 1
262	MOVE.L	0(A3,D6.W),A0	GET PTR TO UDR1 (ARF 1 OR 2)
263	MOVE.B	#TERM,OBUF+52	INSERT TERMINATE
264	MOVE.B	#CR,OBUF+51	INSERT END OF LINE
265	LEA	OBUF+51,A1	SET 1 PAST END OF LINE
266	*		
267	MOVE.W	UDR1_ROL(A0),D2	FORMAT ROLL ANGLE
268	MOVE.W	#TWobytes,D1	
269	JSR	HEXSTR	
270	MOVE.B	#BLNK,-(A1)	
271	*		
272	MOVE.W	UDR1_HEA(A0),D2	FORMAT HEADING
273	MOVE.W	#TWobytes,D1	
274	JSR	HEXSTR	
275	MOVE.B	#BLNK,-(A1)	
276	*		
277	MOVE.W	UDR1_QUA(A0),D2	FORMAT QUALITY
278	MOVE.W	#TWobytes,D1	

Improved GUARDRAIL V MC68000 'DF' Files
 DRCl: [ALGO.IGR.MC68000_DF]DFRPT.S68_NUM

```

279 JSR    HEXSTR
280 MOVE.B #BLNK, -(A1)
281 *
282 MOVE.W UDR1_LOB(A0), D2    FORMAT LOR
283 MOVE.W #TWOBYTES, D1
284 JSR    HEXSTR
285 MOVE.B #BLNK, -(A1)
286 *
287 MOVE.L UDR1_LON(A0), D2    FORMAT LONGITUDE
288 MOVE.W #FOURBYTE, D1
289 JSR    HEXSTR
290 MOVE.B #BLNK, -(A1)
291 *
292 MOVE.L UDR1_LAT(A0), D2    FORMAT LATITUDE
293 MOVE.W #FOURBYTE, D1
294 JSR    HEXSTR
295 MOVE.B #BLNK, -(A1)
296 *
297 MOVE.B UDR1_SSM(A0), D2    FORMAT SIGNAL STRENGTH
298 MOVE.W #ONEBYTE, D1
299 JSR    HEXSTR
300 MOVE.B #BLNK, -(A1)
301 *
302 MOVE.B UDR1_TYP(A0), D2    FORMAT DF TYPE
303 MOVE.W #ONEBYTE, D1
304 JSR    HEXSTR
305 MOVE.B #BLNK, -(A1)
306 *
307 MOVE.W UDR1_ERR(A0), D2    FORMAT ERROR CODE
308 MOVE.W #TWOBYTES, D1
309 JSR    HEXSTR
310 MOVE.B #BLNK, -(A1)
311 *
312 LEA    ARF1ID, A2
313 MOVE.L 0(A2, D6.W), D2
314 MOVE.W #1, D1
315 JSR    HEXSTR
316 MOVE.B #BLNK, -(A1)
317 *
318 LEA    OBUF, A6
319 JSR    WRITEH
320 *
321 JSR    GETPRMPT
322 *
323 MOVE.B #TERM, OBUF+41
324 MOVE.B #CR, OBUF+40
325 LEA    OBUF+40, A1
  
```

GET PTR TO ARF 1 ID.
 GET ARF ID (1 OR 2)
 ONLY ONE DIGIT

GET PTR TO START OF STRING

WAIT FOR A PROMPT

INSERT TERMINATION
 INSERT END OF LINE
 SET PTR TO 1 PAST END

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFRPT.S68_NUM

326	MOVE.W	#NBL,D3	GET MAX NUMBER OF BLS
327	ASL.W	#1,D3	EACH BL IS A WORD
328	LEA	UDR1 BAS(A0),A2	GET PTR TO START OF BLS
329	ADD.W	D3,A2	SET PTR TO 1 PAST LAST BL
330	MOVE.W	#NBL,D3	GET MAX NUMBER
331	SUBQ.W	#1,D3	DECR. FOR COUNTER
332	MOVE.W	#TWOBYTES,D1	SET SIZE
333	*		
334	BLLOOP		
335	MOVE.W	-(A2),D2	GET A BL
336	JSR	HEXSTR	
337	MOVE.B	#BLNK, -(A1)	
338	DBF	D3, BLLOOP	
339	*		
340	LEA	OBUF,A6	SET PTR TO START OF STRING
341	JSR	WRITEH	
342	*		
343	JSR	GETPRMPT	
344	MOVE.L	A0,A6	GET UDR1 PACKET PTR
345	JSR	RELPAK	RELEASE UDR1
346	*		
347	MOVE.L	#NULL,0(A3,D6.W)	INDICATE NO MORE UDR1
348	*		
349	LEA	UDR2ME1,A3	GET ADDR. OF UDR2 ARF 1
350	MOVE.L	0(A3,D6.W),A0	GET UDR2 (ARF 1 OR 2)
351	MOVE.B	#TERM,OBUF+46	INSERT TERMINATION
352	MOVE.B	#CR,OBUF+45	INSERT END OF LINE
353	LEA	OBUF+45,A1	GET PTR TO 1 PAST LINE
354	*		
355	MOVE.W	#5,D3	GET NUMBER OF ELEMENTS WANTED
356	ASL.W	#2,D3	EACH ITEM IS A LONGWORD
357	LEA	UDR2_ER2(A0),A2	GET PTR TO START OF ARRAY
358	ADD.W	D3,A2	SET PTR TO 1 PAST 5TH ITEM
359	MOVE.W	#5,D3	GET NUMBER OF WANTED
360	SUBQ.W	#1,D3	DECR. FOR COUNTER
361	MOVE.W	#FOURBYTE,D1	SET ITEM SIZE
362	*		
363	ERFRLP		
364	MOVE.L	-(A2),D2	GET AN ELEMENT
365	JSR	HEXSTR	FORMAT IT
366	MOVE.B	#BLNK, -(A1)	INSERT A BLANK
367	DBF	D3,ERFRLP	ENDLOOP
368	*		
369	LEA	OBUF,A6	GET PTR TO START OF STRING
370	JSR	WRITEH	WRITE IT OUT
371	*		
372	JSR	GETPRMPT	WAIT FOR A PROMPT

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFRPT.S68_NUM

```

373 MOVE.B #TERM,OBUF+37      INSERT A TERMINATION
374 MOVE.B #CR,OBUF+16       INSERT AN END OF LINE
375 LEA    OBUF+36,A1        GET PTR TO 1 PAST END
376 *
377 MOVE.W #9,D3             GET TOTAL NUMBER OF ITEMS
378 ASL.W #2,D3             EACH ITEM IS A LONGWORD
379 LEA    UDR2,ER2(A0),A2   GET PTR TO START OF ARRAY
380 ADD.W D3,A2             SET PTR TO 1 PAST LAST
381 MOVE.W #4,D3             GET NUMBER WANTED
382 SUBQ.W #1,D3            ADJUST FOR COUNTER
383 MOVE.W #FOURBYTE,D1     SET ITEM SIZE
384 *
385 ERCLP MOVE.L -(A2),D2     GET AN ELEMENT
386 JSR    HEXSTR            FORMAT IT
387 MOVE.B #BLNK, -(A1)      INSERT A BLANK
388 DBF    D3,ERCLP         ENDLOOP
389 *
390 LEA    OBUF,A6           GET PTR TO START OF STRING
391 JSR    WRITE            WRITE IT OUT
392 *
393 JSR    GETPRMPT         WAIT FOR A PROMPT
394 *
395 MOVE.L A0,A6             GET PTR TO UDR2 PACKET
396 JSR    RELPAK          RELEASE UDR2
397 *
398 MOVE.L #NULL,0(A3,D6.W) INDICATE NO MORE UDR2
399 *
400 LEA    UDR3ME1,A3        GET PTR TO UDR3 ARF 1
401 MOVE.L 0(A3,D6.W),A0     GET UDR3 (ARF 1 OR 2)
402 *
403 MOVE.B #TERM,OBUF+37     INSERT TERMINATION
404 MOVE.B #CR,OBUF+36       INSERT END OF LINE
405 LEA    OBUF+36,A1        SET PTR TO 1 PAST END OF LINE
406 *
407 MOVE.W #4,D3             GET NUMBER OF ITEMS WANTED
408 ASL.W #2,D3             EACH ITEM IS A LONGWORD
409 LEA    UDR3,ER2(A0),A2   SET PTR TO START OF ARRAY
410 ADD.W D3,A2             SET PTR TO 1 PAST LAST
411 MOVE.W #4,D3             GET NUMBER WANTED
412 SUBQ.W #1,D3            DECR. FOR COUNTER
413 MOVE.W #FOURBYTE,D1     SET ITEM SIZE
414 *
415 ERTHLP MOVE.L -(A2),D2   GET AN ELEMENT
416 JSR    HEXSTR            FORMAT IT
417 MOVE.B #BLNK, -(A1)      INSERT A BLANK
418 DBF    D3,ERTHLP
419 *

```

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFRPT.S68_NUM

420	LEA	OBUF, A6	SET PTR TO START OF STRING
421	JSR	WRITEH	WRITE IT OUT
422	*		
423	JSR	GETPRMT	WAIT FOR A PROMPT
424	*		
425	MOVE.B	#TERM, OBUF+37	INSERT TERMINATION
426	MOVE.B	#CR, OBUF+36	INSERT END OF LINE
427	LEA	OBUF+36, A1	SET PTR TO 1 PAST LAST
428	*		
429	MOVE.W	UDR3_SCV(A0), D2	FORMAT SEC MIN VALUE
430	MOVE.W	#TWOBYTES, D1	
431	JSR	HEXSTR	
432	MOVE.B	#BLNK, -(A1)	
433	*		
434	MOVE.W	UDR3_SCV(A0), D2	FORMAT SEC MIN INDEX
435	MOVE.W	#TWOBYTES, D1	
436	JSR	HEXSTR	
437	MOVE.B	#BLNK, -(A1)	
438	*		
439	MOVE.W	UDR3_MV1(A0), D2	FORMAT FIRST MIN VALUE
440	MOVE.W	#TWOBYTES, D1	
441	JSR	HEXSTR	
442	MOVE.B	#BLNK, -(A1)	
443	*		
444	MOVE.W	UDR3_MV1(A0), D2	FORMAT FIRST MIN INDEX
445	MOVE.W	#TWOBYTES, D1	
446	JSR	HEXSTR	
447	MOVE.B	#BLNK, -(A1)	
448	*		
449	MOVE.W	UDR3_BAD(A0), D2	FORMAT BAD BL
450	MOVE.W	#TWOBYTES, D1	
451	JSR	HEXSTR	
452	MOVE.B	#BLNK, -(A1)	
453	*		
454	MOVE.B	UDR3_FR5(A0), D2	FORMAT FREQUENCY
455	MOVE.W	#ONEBYTE, D1	
456	JSR	HEXSTR	
457	*		
458	MOVE.B	UDR3_FR4(A0), D2	
459	JSR	HEXSTR	
460	*		
461	MOVE.B	UDR3_FR3(A0), D2	
462	JSR	HEXSTR	
463	*		
464	MOVE.B	UDR3_FR2(A0), D2	
465	JSR	HEXSTR	
466	*		

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1: [ALGO.IGR.MC68000_DF]DFRPT.S68_NUM

467	MOVE.B	UDR3 FR1(A0),D2	
468	JSR	HEXSTR	
469	*		
470	MOVE.B	#BLNK,-(A1)	
471	LEA	OBUFF,A6	GET PTR TO START OF STRING
472	JSR	WRITEH	WRITE STRING OUT
473	*		
474	JSR	GETPRMPT	WAIT FOR A PROMPT
475	*		
476	MOVE.L	A0,A6	GET UDR3 PACKET PTR
477	JSR	RELPAK	RELEASE UDR3
478	*		
479	MOVE.L	#NULL,0(A3,D6.W)	INDICATE NO MORE UDR3
480	*		
481	CLR.W	REPIP	RESET REPORT IN PROGRESS
482	*		
483	DONE	?EXIT	
484		END	

Improved GUARDRAIL V MC68000 'DF' Files

20 FILES WITH EXTENSION '.SCR'

This section contains the files with filenames starting with "DF" and ending in the above extension. Line numbers are added to the right side for ease of reference.

Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000_DF]DFSMENU1.SCR_NUM

20.1 DFSMENU1.SCR_NUM

***** Source Listing ==> DFSMENU1.SCR_NUM *****

```

1 BEGIN
2 LABEL DFSMENU1
3 NEW
4 BODY
5 FDF SIO TEST FOR DPU UNDER TEST
6
7 This test will load a routine from the test set EPROM to
8 the RAM of CPU ' in the DPU under test. That routine will
9 perform the SIO test on the FDF SIO card. This test
10 consists of the following sub-tests:
11 1. SIO channel A interrupt test,
12 2. SIO channel B interrupt test,
13 3. SIO channel C interrupt test,
14 4. SIO channel A internal loop test,
15 5. SIO channel B internal loop test and
16 6. SIO external monitor test.
17 The TEK 318S1 logic analyzer will be used to test the FDF SIO.
18 Configure the TEK 318S1 using NVM setup 3 (see section 9 of
19 "318/338 Operator's Manual" for a description of NVM usage).
20
21 CONNECT THE FOLLOWING CABLES BEFORE PROCEEDING
22
23 CABLE TO FROM CABLE P/N
24 W16 ADPU P2>J12 TEK 318S1 P1>A 10-165316-1
25
26 HIT ANY KEY TO PROCEED
27 .> DFSMENU2 BEGIN
28 .END

```


Improved GUARDRAIL V MC68000 'DF' Files
 DRC1:[ALGO.IGR.MC68000 DF]DFSMENU2.SCR NUM

20.2 DFSMENU2.SCR_NUM

***** Source Listing ==> DF:MENU2.SCR_NUM *****

```

1 BEGIN
2 LABEL DFSMENU2
3 CUR
4 BODY
5
6
7 The FDF SIO differs from the other SIO cards since it
8 does not use the load strobe generated on the SIO card
9 but rather uses the load strobes (one for channel A and
10 one for channel B) provided by the FDF controller card.
11 The FDF controller card monitors the bit clock of the
12 FDF SIO to determine when to proceed to its next state.
13 If the FDF controller does not provide the load strobe to
14 the FDF SIO then the SIO will not initiate a transmission
15 of data. If the FDF SIO does not transmit data when the
16 FDF controller provides the correct load strobes then the
17 FDF controller's state machine will be in a state of waiting
18 for FDF SIO clock pulses which will never appear. If the
19 FDF SIO does not appear to be functioning it will have to
20 be determined if the FDF controller is supplying the correct
21 load strobes before stating that the FDF SIO is faulty.
22
23 HIT ANY KEY TO PROCEED
24 -> DFSMENU3 OR DF:MENU4 OR BUSERR BEGIN
25 END

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000 DF]DFSMENU3.SCR_NUM

20.3 DFSMENU3.SCR_NUM

```
***** Source Listing --> DFSMENU3.SCR_NUM *****
*****
1 .BEGIN
2 .LABEL DFSMENU3
3 .CUR
4 .BODY
5
6 TEST SET DPU NOT ABLE TO READ RESULT OF FDF SIO TEST:
7 FDF SIO CARD (10-156022-1, A13) ASSUMED FAULTY.
8
9 HIT "Q" TO QUIT
10 OR "R" TO REPEAT
11 .END
```

Improved GUARDRAIL V MC68000 'DF' Files
DRCL:[ALGO.IGR.MC68000 DF]DFSMENU4.SCR_NUM

20 4 DFSMENU4.SCR_NUM

```
***** Source Listing ==> DFSMENU4.SCR_NUM *****
*****
1 BEGIN
2 LABEL DFSMENU4
3 CUR
4 BODY
5
6 RESULTS OF PDF I/O TEST
7 channel A interrupt test >
8 LABEL DFSLINE1
9 channel B interrupt test >
10 LABEL DFSLINE2
11 channel C interrupt test >
12 LABEL DFSLINE3
13 channel A internal loop test >
14 LABEL DFSLINE4
15 channel B internal loop test >
16 LABEL DFSLINE5
17
18 EXTERNAL MONITOR TEST IS CURRENTLY RUNNING
19
20 HIT ANY KEY TO PROCEED
21 > DFSMENU4 BEGIN
22 .END
```

20.5 DFSMENU5.SCR_NUM

***** Source Listing ==> DFSMENU5.SCR.NUM *****

```

1 .BEGIN
2 .LABEL DFSMENU5
3 .CUR
4 .BODY
5
6 FDF SIO MONITOR TEST RESULTS
7
8 The bit pattern sent from the FDF SIO should be a 24 bit
9 alternating 1/0 pattern for channel A and a 56 bit
10 alternating 1/0 pattern for channel B. The data will be
11 complemented every cycle. The TEK 318S1 channels are assigned
12 as follows:
13 CHANNEL SIGNAL
14 0 channel A data
15 1 channel A load strobe
16 2 channel A bit clock (24 CYCLES, 250 KHZ)
17 3 channel B data
18 4 channel B load strobe
19 5 channel B bit clock (56 CYCLES, 125 KHZ)
20 Data is valid on the rising edge of the bit clock when the load
21 strobe is low. The clock rate should be 250 khz for channel A and
22 125 khz for channel B. The load strobe should go high at the end
23 of each cycle.
24
25 HIT THE APPROPRIATE KEY TO PROCEED
26 "P" - PASS
27 "F" - FAIL (load strobe goes low for both channels and data
28 clock present but data of either/both channels is incorrect)
29 "O" - OTHER
30 .> DFSMENU6 BEGIN
31 .END

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000 DF]DFSMENU6.SCR_NUM

20.6 DFSMENU6.SCR_NUM

```
***** Source Listing --> DFSMENU6.SCR_NUM *****
*****
1 .BEGIN
2 .LABEL DFSMENU6
3 .CUR
4 .BODY
5
6 FDF SIO / FDF CONTROLLER FAULT ISOLATION
7
8 1. Power down the ADPU by removing P2 of W3 from J15 of the ADPU,
9 2. Remove the ADPU bottom cover to gain access to the backplane,
10 3. Replace P2 of W3 to J15 of the ADPU.
11
12 HIT ANY KEY TO PROCEED
13 .> DFSMENU7 BEGIN
14 .END
```

Improved GUARDRAIL V MC68000 'DF' Files
 DRCL: {ALGO.IGR.MC68000_DF}DFSMENU7.SCR_NUM

20.7 DFSMENU7.SCR_NUM

```

***** Source Listing --> DFSMENU7.SCR_NUM *****
*****
1 BEGIN
2 LABEL DFSMENU7
3 CUR
4 BODY
5
6 INSTRUCTIONS FOR PROBING FDF SIO CARD
7
8 The following table presents the conditions that would
9 imply that the FDF controller card is functioning
10 correctly given that one or both channels of the FDF SIO
11 have failed.
12
13 SIO FDF CONTROLLER FDF CONTROLLER
14 CHANNEL CH A LOAD STROBE CH B LOAD STROBE
15 FAILURE (SLT A13 PIN 20) (SLT A13 PIN 24)
16
17 A HIGH LOW
18 B LOW HIGH
19 A and B HIGH HIGH
20
21 IF THE TWO LOAD STROBE CONDITIONS ABOVE HAVE BEEN MET FOR
22 THE PARTICULAR FDF SIO FAILURE THEN THE FAULT HAS BEEN
23 ISOLATED TO THE FDF SIO AND IT IS CONSIDERED TO HAVE FAILED
24 OTHERWISE THE FDF SIO TEST IS CONSIDERED TO HAVE PASSED.
25
26 HIT THE APPROPRIATE KEY TO PROCEED
27 "P" - PASS (FDF CONTROLLER NOT PROVIDING LOAD STROBE)
28 "F" - FAIL (FDF CONTROLLER IS PROVIDING LOAD STROBE)
29 LABEL DFSLINE6
30
31 HIT "Q" TO QUIT
32 OR "R" TO REPEAT
33 LABEL DFSLINE7
34
35 FDF SIO CARD (1)-156022-1, A11) HAS FAILED
36 LABEL DFSLINE8
37
38 FDF CONTROLLER 10-15850-1, A25) HAS FAILED.
39 END

```

Improved GUARDRAIL V MC68000 'DF' Files
DRC1:[ALGO.IGR.MC68000_DF]DFSMENU8.SCR_NUM

20.8 DFSMENU8.SCR_NUM

```
***** Source Listing ==> DFSMENU8 SCR_NUM *****
*****
1 .BEGIN
2 .LABEL DFSMENU8
3 .CUR
4 .BODY
5
6     HIT "Q" TO QUIT
7     OR "R" TO REPEAT
8 .END
```

INDEX

DF.PAC_NUM, 65	DFF.PAC_NUM, 68
DF.PIC_NUM, 308	DFF.PIC_NUM, 314
DF1.DBS_NUM, 25	DFFEED.R68_NUM, 92
DF1.PIC_NUM, 311	DFFEED.S68_NUM, 371
DFA1.FMT_NUM, 135	DFG1.FMT_NUM, 184
DFA1.LST, 242	DFG1.MOD_NUM, 44
DFA1.MOD_NUM, 39	DFG1.R68_NUM, 93
DFA1.R68_NUM, 77	DFG1.S68_NUM, 373
DFA1.S68_NUM, 324	DFG2.FMT_NUM, 186
DFA2.FMT_NUM, 137	DFG2.MOD_NUM, 45
DFA2.LST, 253	DFG2.R68_NUM, 94
DFA2.MOD_NUM, 40	DFG2.S68_NUM, 378
DFA2.R68_NUM, 78	DFG3.FMT_NUM, 188
DFA2.S68_NUM, 328	DFG3.MOD_NUM, 46
DFA3.FMT_NUM, 140	DFG3.R68_NUM, 95
DFA4.FMT_NUM, 142	DFG3.S68_NUM, 380
DFA4.MOD_NUM, 41	DFG5.FMT_NUM, 190
DFA4.R68_NUM, 82	DFG5.MOD_NUM, 47
DFA4.S68_NUM, 350	DFG5.R68_NUM, 99
DFA5.FMT_NUM, 144	DFG5.S68_NUM, 397
DFA5.MOD_NUM, 42	DFG5T.FMT_NUM, 192
DFA5.R68_NUM, 83	DFG5T.MOD_NUM, 48
DFA5.S68_NUM, 352	DFG5T.R68_NUM, 100
DFA6.FMT_NUM, 146	DFG5T.S68_NUM, 401
DFA6.LST, 285	DFG6.FMT_NUM, 193
DFA6.MOD_NUM, 43	DFG6.MOD_NUM, 49
DFA6.R68_NUM, 85	DFG6.R68_NUM, 101
DFA6.S68_NUM, 357	DFG6.S68_NUM, 403
DFALG.BAT_NUM, 3	DFG7.FMT_NUM, 195
DFALG.CMD_NUM, 5	DFG7.MOD_NUM, 50
DFALG.CNT_NUM, 7	DFG8.FMT_NUM, 197
DFALG.DOC_NUM, 30	DFG8.MOD_NUM, 51
DFALG.FMT_NUM, 148	DFG8.R68_NUM, 102
DFALG.MST_NUM, 60	DFG8.S68_NUM, 406
DFBANDS.D68_NUM, 14	DFG9.FMT_NUM, 199
DFBANDS.R68_NUM, 88	DFG9.MOD_NUM, 52
DFC.DBS_NUM, 27	DFGRND.CNT_NUM, 10
DFC.PIC_NUM, 313	DFGRND.DOC_NUM, 31
DFCOL.D68_NUM, 15	DFGRND.FMT_NUM, 201
DFCOL.R68_NUM, 89	DFGRND.MST_NUM, 62
DFCOL2.D68_NUM, 19	DFI.PAC_NUM, 70
DFCOL2.R68_NUM, 90	DFIN.FMT_NUM, 232
DFCOM.D68_NUM, 23	DFIN.MOD_NUM, 53
DFCOM.R68_NUM, 91	DFIN.R68_NUM, 103
DFERR.INC_NUM, 36	DFIN.S68_NUM, 410
DFERRS.EQU_NUM, 33	DFOUT.CSS_NUM, 12

DFOUT.FMT_NUM, 234
 DFOUT.MOD_NUM, 54
 DFOUT.R68_NUM, 104
 DFOUT.S68_NUM, 412
 DFPE1.FMT_NUM, 236
 DFPE1.MOD_NUM, 56
 DFPE1.R68_NUM, 105
 DFPE1.S68_NUM, 416
 DFPE2.FMT_NUM, 238
 DFPE2.MOD_NUM, 57
 DFPE2.R68_NUM, 106
 DFPE2.S68_NUM, 418
 DFPON.FMT_NUM, 240
 DFPON.MOD_NUM, 58
 DFPON.R68_NUM, 108
 DFPON.S68_NUM, 428
 DFPR.PAC_NUM, 71
 DFPR.PIC_NUM, 315
 DFQ.DBS_NUM, 28
 DFQ.PIC_NUM, 318
 DFR.PAC_NUM, 73
 DFR.PIC_NUM, 319
 DFRJ.PAC_NUM, 75
 DFRJ.PIC_NUM, 322
 DFRPT.R68_NUM, 109
 DFRPT.S68_NUM, 431
 DFSMENU1.DEF_NUM, 113
 DFSMENU1.SCR_NUM, 443
 DFSMENU2.DEF_NUM, 117
 DFSMENU2.SCR_NUM, 444
 DFSMENU3.DEF_NUM, 120
 DFSMENU3.SCR_NUM, 445
 DFSMENU4.DEF_NUM, 121

DFSMENU4.SCR_NUM, 446
 DFSMENU5.DEF_NUM, 123
 DFSMENU5.SCR_NUM, 447
 DFSMENU6.DEF_NUM, 127
 DFSMENU6.SCR_NUM, 448
 DFSMENU7.DEF_NUM, 129
 DFSMENU7.SCR_NUM, 449
 DFSMENU8.DEF_NUM, 133
 DFSMENU8.SCR_NUM, 450
 DFTYPE.EQU_NUM, 34
 DFTYPE.INC_NUM, 37

FILES WITH EXTENSION '.BAT', 2
 FILES WITH EXTENSION '.CMD', 4
 FILES WITH EXTENSION '.CNT', 6
 FILES WITH EXTENSION '.CSS', 11
 FILES WITH EXTENSION '.D68', 13
 FILES WITH EXTENSION '.DBS', 24
 FILES WITH EXTENSION '.DEF', 112
 FILES WITH EXTENSION '.DOC', 29
 FILES WITH EXTENSION '.EQU', 32
 FILES WITH EXTENSION '.FMT', 134
 FILES WITH EXTENSION '.INC', 35
 FILES WITH EXTENSION '.LST', 241
 FILES WITH EXTENSION '.MOD', 38
 FILES WITH EXTENSION '.MST', 59
 FILES WITH EXTENSION '.PAC', 64
 FILES WITH EXTENSION '.PIC', 307
 FILES WITH EXTENSION '.R68', 76
 FILES WITH EXTENSION '.S68', 323
 FILES WITH EXTENSION '.SCR', 442

Introduction, 1

END

10-87

DTIC